

The `physicx` package

Wenjian Chern (Longaster*)

October 22, 2022, version v0.3.5

Abstract

`physicx`

1 Implementation

```
1 <*package>
2 <@@=physicx>
3 \cs_generate_variant:Nn \keys_set:nn { nx , on , ox }
4 \cs_generate_variant:Nn \use:n nnn { nnno }
5 \cs_generate_variant:Nn \seq_set_split:Nnn { Non, NVV, c, cnV, cVV }
6 \cs_generate_variant:Nn \tl_replace_all:Nnn { Non, Nox }
7 \cs_new:Npn \PHYSICXIGNORE
8 { \exp_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

```

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 [\+\\-\\ ]*(\d+|\d*\.\d+) \Z } {#1}
178       { \prg_return_true: } { \prg_return_false: }
179     }
180   \cs_new:Npn \physicx_search_also:nn #1#2
181     {
182       \clist_map_inline:nn {#1}
183       {
184         \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
185         {
186           \clist_map_break:n
187           { \keys_set:no {##1} { \l_keys_key_str = {#2} } }
188         }
189       }

```

```

189     }
190 }
191 \prg_new_conditional:Npnn \physicsx_search_also:nn #1#2 { T, F, TF }
192 {
193   \bool_set_false:N \l__physicsx_tmpa_bool
194   \clist_map_inline:nn {#1}
195   {
196     \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
197     {
198       \clist_map_break:n
199       {
200         \bool_set_true:N \l__physicsx_tmpa_bool
201         \keys_set:no {##1} { \l_keys_key_str = {#2} }
202       }
203     }
204   }
205   \bool_if:NTF \l__physicsx_tmpa_bool
206   { \prg_return_true: } { \prg_return_false: }
207 }
208 \cs_generate_variant:Nn \physicsx_search_also:nn { no , oo }
209 \prg_generate_conditional_variant:Nnn \physicsx_search_also:nn { no , oo } { T , F , TF }
210 \cs_new_protected:Npn \physicsx_new_type:nnn #1#2#3
211 { \keys_define:nn { physicsx/#1 } { type / #2 .meta:n = {#3} } }
212 \tl_const:Nn \c_physicsx_order_tl { \mathcal{o} }
213 \tl_const:Nn \c_physicsx_Order_tl { \mathcal{O} }
214 \cs_new:Npn \physicsx_use_amssymb_type:
215 {
216   \cs_set_eq:NN \physicsx_bf: \boldsymbol
217 }
218 \cs_new:Npn \physicsx_use_uni_bfit_type:
219 {
220   \cs_set_eq:NN \physicsx_bf: \symbfit
221 }
222 \cs_new:Npn \physicsx_use_uni_bf_type:
223 {
224   \cs_set_eq:NN \physicsx_bf: \symbf
225 }
226 \cs_new:Npn \physicsx_left: { \mathopen{}\mathclose\bgroup\left }
227 \cs_new:Npn \physicsx_right: { \aftergroup\egroup\right }
228 \cs_new:Npn \physicsx_left:N { \mathopen{}\mathclose\bgroup }
229 \cs_new:Npn \physicsx_right:N { \egroup }
230 \cs_new_protected_nopar:Npn \physicsx_left:nN
231 { \__physicsx_delsize:NNnn \physicsx_left: \tex_mathopen:D }
232 \cs_new_protected_nopar:Npn \physicsx_right:nN
233 { \__physicsx_delsize:NNnn \physicsx_right: \tex_mathclose:D }
234 \cs_new_nopar:Npn \__physicsx_delsize:NNnn #1#2#3#4
235 {
236   \fp_compare:nNnTF {#3} < { 0 }
237   { #1 #4 }
238   { #2 { \exp_args:Nf \bBigg@ { \fp_eval:n { (#3)/1.2 } } } {#4} } }
239 }
240 \cs_new:Npn \__physicsx_loadpackage_options:nnn #1#2#3
241 {

```

```

242     \clist_if_empty:nF {#1} { \PassOptionsToPackage {#1} {#3} }
243     \RequirePackage {#3}
244 }
245 \keys_define:nn { physicsx }
246 {
247     compat .bool_set:N = \g__physicsx_compat_bool ,
248     compat .default:n = true ,
249     short .bool_set:N = \g__physicsx_short_bool ,
250     short .default:n = true ,
251     physics .code:n = \__physicsx_loadpackage_options:nnn {#1} { } {physics} ,
252     physics .default:n = { } ,
253     mathtools .code:n = \__physicsx_loadpackage_options:nnn {#1} { } {mathtools} ,
254     mathtools .default:n = { } ,
255     unimath .code:n = \__physicsx_loadpackage_options:nnn {#1} { } { unicode-math } ,
256     unimath .default:n = { } ,
257     reqty .bool_set:N = \g__physicsx_reqty_bool ,
258     reqty .default:n = true ,
259     reqty .initial:n = true ,
260     noqty .meta:n = { reqty = false } ,
261     fixdif .bool_set:N = \g__physicsx_fixdif_bool ,
262     original .bool_set:N = \g__physicsx_original_bool ,
263 }
264 %
265 \ProcessKeysPackageOptions { physicsx }
266 %
267 \@ifpackageloaded{physics}
268 { \bool_set_true:N \g__physicsx_compat_bool }
269 {
270     \bool_if:NT \g__physicsx_compat_bool
271     {
272         \AtBeginDocument
273         {
274             \cs_set_eq:NN \divisionsymbol \div
275             \cs_set_eq:NN \real \Re
276             \cs_set_eq:NN \imaginary \Im
277         }
278     }
279 }
280 \@ifpackageloaded{mathtools}
281 { \bool_set_true:N \g__physicsx_mathtools_bool }
282 { \bool_set_false:N \g__physicsx_mathtools_bool }
283 %
284 \physicsx_compat:T
285 {
286     \tl_set_eq:NN \ordersymbol \c_physicsx_order_tl
287     \tl_set_eq:NN \Ordersymbol \c_physicsx_Order_tl
288 }
289 %
290 \@ifpackageloaded {unicode-math}
291 { \physicsx_use_uni_bfit_type: }
292 { \physicsx_use_amssymb_type: }
293 \physicsx_unimath:T { %% TODO:
294     \cs_set:Npn \__physicsx_vnabla: { \sympbf \nabla }
295     \AtBeginDocument{

```

```

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

```

```

350     \tl_if_head_eq_meaning:nNTF {#2} *
351     { \exp_args:NNc #1 * { \tl_tail:n {#2} } {#3} }
352     { \exp_args:Nc #1 {#2} {#3} }
353 }
354 \cs_new_protected:Npn \__physicx_fixdif:Nn #1#2
355 {
356     \exp_args:NNnx \__physicx_fixdif:Nnn #1 {#2}
357     { \tl_if_head_eq_meaning:nNTF {#2} * { \tl_tail:n {#2} } {#2} }
358 }

```

`\physicxset` `physicx` setup command.

```

359 \NewDocumentCommand \physicxset { s m }
360 {
361     \IfBooleanTF {#1}
362     { \keys_set:nn { physicx/#2 } }
363     { \keys_set:nn { physicx } {#2} }
364 }

```

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

1.2 Quantity things

1.2.1 New quantity interfaces

```

365 \tl_new:N \l__physicx_quantity_args_tl
366 \tl_new:N \l__physicx_quantity_code_tl
367 \tl_new:N \l__physicx_quantity_left_size_tl
368 \tl_new:N \l__physicx_quantity_left_tl
369 \tl_new:N \l__physicx_quantity_post_tl
370 \tl_new:N \l__physicx_quantity_pre_tl
371 \tl_new:N \l__physicx_quantity_right_size_tl
372 \tl_new:N \l__physicx_quantity_right_tl
373 \keys_define:nn { physicx }
374 { quantity .code:n = \keys_set:nn { physicx/quantity } {#1} }
375 \keys_define:nn { physicx/quantity }
376 {
377     pre .tl_set:N = \l__physicx_quantity_pre_tl ,
378     post .tl_set:N = \l__physicx_quantity_post_tl ,
379     left .tl_set:N = \l__physicx_quantity_left_tl ,
380     right .tl_set:N = \l__physicx_quantity_right_tl ,
381     left-size .code:n = { \__physicx_quantity_size:nn { left } {#1} } ,
382     right-size .code:n = { \__physicx_quantity_size:nn { right } {#1} } ,
383     size .meta:n = { left-size = {#1} , right-size = {#1} } ,
384     noauto .meta:n = { left-size = \c_empty_tl , right-size = \c_empty_tl } ,
385     noauto .value_required:n = false ,
386     args .code:n =
387     { \tl_set:Nn \l__physicx_quantity_args_tl { [#1] } ,
388     args* .tl_set:N = \l__physicx_quantity_args_tl ,
389     code .tl_set:N = \l__physicx_quantity_code_tl ,
390     type .multichoice: ,
391
392     settype .code:n = \setquantitytype #1 ,
393
394     unknown .code:n = \__physicx_quantity_unknown:n {#1} ,

```



```

395 }
396 \cs_new:Npn \__physicx_quantity_size:nn #1#2
397 {
398   \physicx_if_num:nTF {#2}
399   {
400     \tl_set:cx { l__physicx_quantity_ #1 _size_tl }
401     { \use:c { physicx_ #1:nN } { \fp_eval:n {#2} } }
402   }
403   { \tl_set_eq:cN { l__physicx_quantity_ #1 _size_tl } #2 }
404 }
405 \cs_new:Npn \__physicx_quantity_unknown:n #1
406 {
407   \tl_set:Nx \l__physicx_tmpa_tl { \tl_head:N \l_keys_key_str }
408   \token_if_eq_meaning:NNTF \l__physicx_tmpa_tl \c_backslash_str
409   { \use:n } { \use_ii:nn }
410   {
411     \cs_if_exist:cTF { \tl_tail:N \l_keys_key_str }
412     {
413       \keys_set:nx { physicx/quantity }
414       { size = \exp_not:c { \tl_tail:N \l_keys_key_str } }
415       \use_none:n
416     }
417     { \use:n }
418   }
419   {
420     \exp_args:No \physicx_if_num:nTF \l_keys_key_str
421     {
422       \keys_set:nx { physicx/quantity } { size = \l_keys_key_str }
423     }
424     {
425       \physicx_search_also:nnF
426       {
427         physicx/quantity/type ,
428       }
429       {#1}
430       {
431         \msg_error:nxxx { physicx } { unknown-key }
432         \l_keys_path_str { physicx/quantity }
433       }
434     }
435   }
436 }
437 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
438 { \physicx_new_type:nnn { quantity } {#1} }
439 \setquantitytype { b } { left={[] , right={[] } , }
440 \setquantitytype { B } { left={\{ } , right={\} } , }
441 \setquantitytype { p } { left={ ( } , right={ ) } , }
442 \setquantitytype { v } { left=\lvert , right=\rvert , }
443 \setquantitytype { V } { left=\lVert , right=\rVert , }
444 \setquantitytype { a } { left=\langle , right=\rangle , }
445 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
446 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
447 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
448 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }

```

```

449 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
450 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
451 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
452 \physicx_mathtools:T
453 {
454   \setquantitytype { m* } { left=\begin{matrix*} , right=\end{matrix*} , noauto }
455   \setquantitytype { bm* } { left=\begin{bmatrix*} , right=\end{bmatrix*} , noauto }
456   \setquantitytype { Bm* } { left=\begin{Bmatrix*} , right=\end{Bmatrix*} , noauto }
457   \setquantitytype { pm* } { left=\begin{pmatrix*} , right=\end{pmatrix*} , noauto }
458   \setquantitytype { vm* } { left=\begin{vmatrix*} , right=\end{vmatrix*} , noauto }
459   \setquantitytype { Vm* } { left=\begin{Vmatrix*} , right=\end{Vmatrix*} , noauto }
460   \setquantitytype { sm* } { left=\begin{smallmatrix*} , right=\end{smallmatrix*} , noauto }
461   \setquantitytype { sbm* } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
462   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
463   \setquantitytype { spm* } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
464   \setquantitytype { svm* } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
465   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
466   \setquantitytype { sbm* } { left=\begin{bsmallmatrix*} , right=\end{bsmallmatrix*} , noauto }
467   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix*} , right=\end{Bsmallmatrix*} , noauto }
468   \setquantitytype { spm* } { left=\begin{psmallmatrix*} , right=\end{psmallmatrix*} , noauto }
469   \setquantitytype { svm* } { left=\begin{vsmallmatrix*} , right=\end{vsmallmatrix*} , noauto }
470   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix*} , right=\end{Vsmallmatrix*} , noauto }
471 }
472 \keys_set:nn { physicx/quantity }
473 {
474   left-size = \left ,
475   right-size = \right ,
476   type = p ,
477 }

\physicx_xquantity:nn
  \newxquantity
  \NewXQuantity
478 \cs_new:Npn \physicx_xquantity:nn #1#2
479 {
480   \group_begin:
481   \keys_set:nn { physicx/quantity } {#1}
482   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
483   \__physicx_xquantity_aux:oooo
484   { \l__physicx_quantity_left_tl }
485   { \l__physicx_quantity_args_tl }
486   { \l__physicx_quantity_code_tl }
487   { \l__physicx_quantity_right_tl }
488   \group_end:
489 }
490 \cs_new:Npn \__physicx_xquantity_aux:nnnn #1#2#3#4
491 {
492   \l__physicx_quantity_pre_tl
493   \bool_lazy_or:nnTF
494   { \tl_if_empty_p:N \l__physicx_quantity_left_size_tl }
495   { \tl_if_empty_p:N \l__physicx_quantity_right_size_tl }
496   { #1 #2 #3 #4 }
497   {
498     \bool_lazy_or:nnTF
499     { \token_if_eq_meaning_p:NN \l__physicx_quantity_left_size_tl \left }
500     { \token_if_eq_meaning_p:NN \l__physicx_quantity_right_size_tl \right }

```

```

501     { \physicx_left: #1 #2 #3 \physicx_right: #4 }
502     {
503       \exp_args:No \tl_if_head_eq_meaning:nNTF
504       \l__physicx_quantity_left_size_tl \physicx_left:nN
505       {
506         \l__physicx_quantity_left_size_tl #1 #2
507         #3
508         \l__physicx_quantity_right_size_tl #4
509       }
510       {
511         \physicx_left:N \l__physicx_quantity_left_size_tl #1 #2
512         #3
513         \physicx_right:N \l__physicx_quantity_right_size_tl #4
514       }
515     }
516   }
517   \l__physicx_quantity_post_tl
518 }
519 \NewDocumentCommand \xquantity { } { \physicx_xquantity:nn }
520 \cs_generate_variant:Nn \__physicx_xquantity_aux:nnnn { oooo }
521 \NewDocumentCommand \newxquantity { m o o m m }
522 {
523   \IfNoValueTF {#2}
524   {
525     \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
526     { \newcommand ##1 }
527   }
528   {
529     \IfNoValueTF {#3}
530     {
531       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
532       { \newcommand ##1 [#2] }
533     }
534     {
535       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
536       { \newcommand ##1 [#2] [#3] }
537     }
538   }
539   \exp_args:Nc \__physicx_new_xquantity_aux:w
540   { \cs_to_str:N #1~star }
541   { \physicx_xquantity:nn { #4 , noauto } {#5} }
542   \exp_args:Nc \__physicx_new_xquantity_aux:w
543   { \cs_to_str:N #1~unstar }
544   { \physicx_xquantity:nn { #4 } {#5} }
545   \exp_args:NNx \newcommand #1
546   {
547     \exp_not:N \@ifstar
548     \exp_not:c { \cs_to_str:N #1~star }
549     \exp_not:c { \cs_to_str:N #1~unstar }
550   }
551 }
552 \NewDocumentCommand \NewXQuantity { m m m m m }
553 {
554   \NewDocumentCommand #1 { s #2 }

```

```

555     {
556       \IfBooleanTF {##1}
557       { \physicx_xquantity:nn { #3 , noauto } {#4} }
558       { \physicx_xquantity:nn { #3 } {#4} }
559     }
560   }
561   \NewXQuantity \qxqty { 0{ } m } { #2 } {#3}

```

(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
562 \tl_new:N \physicx_tmp
563 \tl_new:N \l__physicx_cmd_noauto_body_tl
564 \bool_new:N \l__physicx_cmd_noauto_body_bool
565 \tl_new:N \l__physicx_cmd_auto_body_tl
566 \bool_new:N \l__physicx_cmd_auto_body_bool
567 \tl_new:N \l__physicx_cmd_arg_spec_tl
568 \int_new:N \l__physicx_cmd_arg_int
569 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
570 {
571   \tl_clear:N \l__physicx_cmd_noauto_body_tl
572   \tl_clear:N \l__physicx_cmd_auto_body_tl
573   \tl_clear:N \l__physicx_cmd_arg_spec_tl
574   \int_set:Nn \l__physicx_cmd_arg_int {#1}
575   \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
576   \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
577 }
578 % noauto, auto, cmd, body
579 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
580 {
581   \__physicx_declare_init:nnn { 3 } {#1} {#2}
582   \__physicx_declare_legacy_quantity_aux:nw #4
583   \q_recursion_tail \q_recursion_tail \q_recursion_stop
584   \__physicx_declare_legacy_quantity_aux:NcVVV
585   #3 { \cs_to_str:N #3 ~ body }
586   \l__physicx_cmd_arg_spec_tl
587   \l__physicx_cmd_noauto_body_tl
588   \l__physicx_cmd_auto_body_tl
589 }
590 % arg spec, pre, body to replace(start from #4), post
591 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
592 {
593   \int_incr:N \l__physicx_cmd_arg_int
594   \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
595     \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
596     \tl_set:Nx \l__physicx_tmp_tl
597     {
598       {
599         \exp_not:N \tl_if_novalue_p:n
600         {
601           \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:

```

```

602         \or: \or: \or:
603         \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
604         \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
605         \fi:
606     }
607 }
608 }
609 \if_bool:N \l__physicx_cmd_noauto_body_bool
610     \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
611     \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
612     {
613         {
614             % if is '.', use none
615             \str_if_eq:nnTF {#2} {.} {} {#2}
616             #3
617             \str_if_eq:nnTF {#4} {.} {} {#4}
618         }
619     }
620 \fi:
621 \if_bool:N \l__physicx_cmd_auto_body_bool
622     \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
623     \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
624     { { ##1 #2 #3 ##2 #4 } }
625 \fi:
626 \fi:
627 }
628 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
629 {
630     \quark_if_recursion_tail_stop:n {#1}
631     \quark_if_recursion_tail_stop:n {#2}
632     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
633     \__physicx_declare_legacy_quantity_aux:nw
634 }
635 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
636 {
637     \__physicx_nauto_case:nnnn
638     { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
639     {
640         \cs_set_protected:Npn #1
641         {
642             \peek_charcode_ignore_spaces:NTF \let
643             { #2 } { #2 [ \physicx_left: ] \physicx_right: }
644         }
645         \DeclareDocumentCommand #2 { 0{##2} m s #3 }
646         {
647             \IfBooleanTF { ##3 }
648             { \bool_case_false:n {#4} }
649             { \bool_case_false:n {#5} }
650         }
651     }
652     {
653         \cs_set_protected:Npn #1
654         { #2 \c_empty_tl \c_empty_tl }
655         \DeclareDocumentCommand #2 { m m s #3 }

```

```

656         { \bool_case_false:n {#4} }
657     }
658 }
659 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
660 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
661 {
662     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
663     {
664         \bool_if:NTF \l__physicx_cmd_auto_body_bool
665         {#1} {#2}
666     }
667     {
668         \bool_if:NTF \l__physicx_cmd_auto_body_bool
669         {#3} {#4}
670     }
671 }
672 \cs_set_protected:Npn \@declarequantitycmd
673 { \physicx_declare_legacy_quantity:nnNn }

```

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

```

\quantity      Redefine some macros in physics package.
\evaluated
\matrixquantity
\smallmatrixquantity
674 \if_bool:N \g__physicx_reqty_bool
675 \physicx_declare_legacy_quantity:nnNn
676 \c_true_bool \c_true_bool \quantity
677 {
678     { !g } { { \{ } { #4 } { \} } }
679     { !o } { { [ } { #5 } { ] } }
680     { !d() } { { ( } { #6 } { ) } }
681     { !d|| } { { \vert } { #7 } { \vert } }
682     { !d<> } { { \langle } { #8 } { \rangle } }
683     { !d== } { { \Vert } { #9 } { \Vert } }
684 }
685 \physicx_declare_legacy_quantity:nnNn
686 \c_true_bool \c_true_bool \evaluated
687 {
688     { !g } { { . } { #4 \nobreak } { \vert } }
689     { !d[ ] } { { [ ] } { #5 \nobreak } { \vert } }
690     { !d( ) } { { ( ) } { #6 \nobreak } { \vert } }
691 }
692 \physicx_declare_legacy_quantity:nnNn
693 \c_true_bool \c_false_bool \matrixquantity
694 {
695     { !g }
696     {
697         { \IfBooleanT{#3}{\left\{ } }
698         { \begin{matrix} #4 \end{matrix} }
699         { \IfBooleanT{#3}{\right\} }
700     }
701     { !o } { { \begin{bmatrix} } {#5} { \end{bmatrix} } }
702     { !d() }
703     {
704         { \IfBooleanTF{#3}{\left\lgroup}{\left( } }

```

```

705     { \begin{matrix} #6 \end{matrix} }
706     { \IfBooleanTF{#3}{\right\rgroup}{\right}} }
707   }
708   { !d|| } { { \begin{vmatrix} } {#7} { \end{vmatrix} } } }
709   { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } } { \right\rangle } }
710   { !d== } { { \begin{Vmatrix} } {#9} { \end{Vmatrix} } } }
711 }
712 \physicsx_declare_legacy_quantity:nnNn
713 \c_true_bool \c_false_bool \smallmatrixquantity
714 {
715   { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } } { \right\} } }
716   { !o } { { \left[ } { \begin{smallmatrix} #5 \end{smallmatrix} } } { \right] } }
717   { !d() }
718   {
719     { \IfBooleanTF{#3}{\left\lgroup}{\left( }
720     { \begin{smallmatrix} #6 \end{smallmatrix} }
721     { \IfBooleanTF{#3}{\right\rgroup}{\right)}} }
722   }
723   { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } } { \right\vert } }
724   { !d<> } { { \left\langle } { \begin{smallmatrix} #8 \end{smallmatrix} } } { \right\rangle } }
725   { !d== } { { \left\Vert } { \begin{smallmatrix} #9 \end{smallmatrix} } } { \right\Vert } }
726 }
727 \fi:

```

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

```

\physicsx_declare_legacy_paren:NnnnNn
  \@declareparenccmd
728 %% cmd, arg spec, replace(start from #6), pre, left, right, post
729 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNn #1#2#3#4#5#6#7
730 {
731   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
732   {
733     \bool_case_true:nF
734     {
735       { \bool_if_p:n {##2} } { #4 \physicsx_left:N \bigl #5 #3 \physicsx_right:N \bigr
736       { \bool_if_p:n {##3} } { #4 \physicsx_left:N \Bigl #5 #3 \physicsx_right:N \Bigr
737       { \bool_if_p:n {##4} } { #4 \physicsx_left:N \biggl #5 #3 \physicsx_right:N \biggr
738       { \bool_if_p:n {##5} } { #4 \physicsx_left:N \Biggl #5 #3 \physicsx_right:N \Biggr
739     }
740     {
741       \IfBooleanTF {##1}
742       { #4 #5 #3 #6 #7 }
743       { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
744     }
745   }
746 }
747 \cs_set_protected:Npn \@declareparenccmd
748 { \physicsx_declare_legacy_paren:NnnnNn }

```

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

```

\qty Redefine some macros in physics package.
\mqty
749 \if_bool:N \g__physicsx_reqty_bool
\smqty \physicsx_option_or:nnT { compat } { short }
\pqty
\bqty
\vqty
\Bqty
\absolutevalue
\eval
\abs
\norm
\order
\oorder

```

```

751 {
752   \cs_set:Npn \qty { \quantity }
753   \physics_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
754   \physics_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
755   \physics_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
756   \physics_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
757 }
758 \physics_declare_legacy_paren:NnnnNNn \absolutevalue
759 { m } {#6} { } { } \vert \vert { }
760 \physics_option_or:nnT { compat } { short }
761 {
762   \cs_set:Npn \eval { \evaluated }
763   \cs_set:Npn \abs { \absolutevalue }
764 }
765 \physics_declare_legacy_paren:NnnnNNn \norm
766 { m } {#6} { } { } \lVert \rVert { }
767 \physics_compat:TF
768 {
769   \physics_declare_legacy_paren:NnnnNNn \order
770   { m } {#6} { } { } \c_physicx_Order_tl { } { }
771 }
772 {
773   \physics_declare_legacy_paren:NnnnNNn \order
774   { m } {#6} { } { } \c_physicx_order_tl { } { }
775 }
776 \physics_declare_legacy_paren:NnnnNNn \commutator
777 { m m } { #6 , #7 } { } { } [ ] { }
778 \physics_option_or:nnT { compat } { short }
779 { \cs_set:Npn \comm { \commutator } }
780 \physics_declare_legacy_paren:NnnnNNn \poissonbracket
781 { m m } { #6 , #7 } { } { } \{ \} { }
782 \physics_option_or:nnT { compat } { short }
783 {
784   \cs_set:Npn \pb { \poissonbracket }
785   \cs_set:Npn \anticommutator { \poissonbracket }
786   \cs_set:Npn \acomm { \poissonbracket }
787 }
788 \fi:
789 \physics_declare_legacy_paren:NnnnNNn \OOrder
790 { m } {#6} { } { } \c_physicx_Order_tl { } { }
791 \physics_declare_legacy_paren:NnnnNNn \oorder
792 { 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

```

793 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
794 {
795   \int_set:Nn \l__physicx_matrix_rows_int
796   { \int_max:nn {#1} \l__physicx_matrix_rows_int }
797   \int_set:Nn \l__physicx_matrix_cols_int
798   { \int_max:nn {#2} \l__physicx_matrix_cols_int }

```



```

799   }
800   % use matrix element
801   \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
802   {
803     \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
804     \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
805     \else:
806     \exp_not:o { \physicxempty }
807     \fi:
808   }
809   % set matrix element, check or not
810   \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
811   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
812   \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
813   {
814     \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
815     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
816   }
817   \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
818   {
819     \tl_if_empty:nTF {#3}
820     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
821     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
822   }
823   \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
824   {
825     \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
826     {
827       \tl_if_empty:nTF {#3}
828       { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
829       { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
830     }
831   }
832   \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
833   \__physicx_matrix_set_r_c_ckigep:nnn
834   \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
835   \__physicx_matrix_set_r_c_nock:nnn
836   % align, cr, sep symbol
837   \str_const:Nn \physicx@align { , }
838   \str_const:Nn \physicx@cr { ; }
839   \str_const:Nn \physicx@sep { , }
840   \bool_new:N \l__physicx_matrix_infinite_bool
841   \bool_new:N \l__physicx_matrix_dotrow_bool
842   \bool_new:N \l__physicx_matrix_dotcol_bool
843   \tl_new:N \l__physicx_matrix_array_tl
844   \tl_new:N \l__physicx_matrix_body_tl
845   \int_new:N \l__physicx_matrix_rows_int
846   \int_new:N \l__physicx_matrix_cols_int
847   \tl_new:N \l__physicx_matrix_main_tl
848   \clist_new:N \l__physicx_matrix_diag_clist
849   \clist_new:N \l__physicx_matrix_item_clist
850   \bool_new:N \l__physicx_matrix_diag_bool
851   \seq_new:N \l__physicx_row_list_seq
852   \seq_new:N \l__physicx_col_list_seq

```

```

853 % expand input
854 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
855 %% main, row iterate, col iterate
856 \cs_new_nopar:Npn \physicx@matricxelement #1#2#3 { #1 \sb { #2 #3 } }
857 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
858 \tl_new:N \l__physicx_matrix_last_row_tl
859 \tl_new:N \l__physicx_matrix_last_col_tl
860 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
861 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
862 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
863 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn
864 \bool_new:N \l__physicx_matrix_expand_element_bool
865 % when element is empty use \physicxempty
866 \tl_new:N \physicxempty
867 % save 'element-except' key's value
868 \tl_new:N \physicxexcept
869 \tl_new:N \l__physicx_matrix_args_tl
870 \tl_new:N \l__physicx_matrix_after_begin_tl
871 \tl_new:N \l__physicx_matrix_after_end_tl
872 \bool_new:N \l__physicx_matrix_transpose_bool
873 \bool_new:N \l__physicx_matrix_enhanced_bool
874 \dim_new:N \l__physicx_matrix_sep_dim
875 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
876 \tl_new:N \l__physicx_matrix_beginning_tl
877 \tl_new:N \l__physicx_matrix_ending_tl

```

1.3.2 Matrix keys

```

878 \keys_define:nn { physicx }
879 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
880 \keys_define:nn { physicx/matrix }
881 {
882   array .tl_set:N = \l__physicx_matrix_array_tl ,
883   expand .choice: ,
884   expand / none .code:n =
885     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
886   expand / text-expand .code:n =
887     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
888   expand / f .code:n =
889     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
890   expand / romanual .meta:n = { expand = f } ,
891   expand / x .code:n =
892     \cs_set_eq:NN \__physicx_expand:w \use:n ,
893   expand / edef .meta:n = { expand = x } ,
894   rows .int_set:N = \l__physicx_matrix_rows_int ,
895   cols .int_set:N = \l__physicx_matrix_cols_int ,
896   auto-update .choice: ,
897   auto-update / true .code:n =
898     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
899   auto-update / false .code:n =
900     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
901   auto-update .default:n = true ,
902   main .tl_set:N = \l__physicx_matrix_main_tl ,
903   row-list .code:n =
904     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,

```

```

905 col-list .code:n =
906   \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
907   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
908   infinite .default:n = true ,
909   !infinite .code:n =
910     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
911   element-code .cs_set:Np = \physicx@matricelement #1#2#3 ,
912   element-code* .choice: ,
913   element-code* / except-empty .code:n =
914     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
915     \physicx@matricelement
916     \cs_set:Npn \physicx@matricelement ##1##2##3
917     {
918       \tl_if_empty:nTF {##1}
919       {##1}
920       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
921     } ,
922   element-code* / except-blank .code:n =
923     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
924     \physicx@matricelement
925     \cs_set:Npn \physicx@matricelement ##1##2##3
926     {
927       \tl_if_blank:nTF {##1}
928       {##1}
929       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
930     } ,
931   element-code* / except-dots .code:n =
932     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
933     \physicx@matricelement
934     \cs_set:Npn \physicx@matricelement ##1##2##3
935     {
936       \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
937       {##1}
938       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
939     } ,
940   element-code* / except-tl .code:n =
941     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
942     \physicx@matricelement
943     \cs_set:Npn \physicx@matricelement ##1##2##3
944     {
945       \tl_if_in:onTF { \physicxexcept } {##1}
946       {##1}
947       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
948     } ,
949   element-code* / except-regex .code:n =
950     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
951     \physicx@matricelement
952     \cs_set:Npn \physicx@matricelement ##1##2##3
953     {
954       \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
955       {##1}
956       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
957     } ,
958   element-code* / only-regex .code:n =

```

```

959 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
960 \physicx@matricelement
961 \cs_set:Npn \physicx@matricelement ##1##2##3
962 {
963   \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
964   { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
965   {##1}
966 } ,
967 element-code* / unknown .code:n =
968   \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} },
969 element-exception .tl_set:N = \physicxexcept ,
970 element-exception+ .code:n =
971   \tl_put_right:Nn \physicxexcept {#1} ,
972 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
973 expand-element .default:n = true ,
974 empty .tl_set:N = \physicxempty ,
975 check .choice: ,
976 check / none .code:n =
977   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
978   \__physicx_matrix_set_r_c_nock:nnn ,
979 check / empty .code:n =
980   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
981   \__physicx_matrix_set_r_c_cke:nnn ,
982 check / ignore .code:n =
983   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
984   \__physicx_matrix_set_r_c_ckig:nnn ,
985 check / igep .code:n =
986   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
987   \__physicx_matrix_set_r_c_ckigep:nnn ,
988 check / all .code:n =
989   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
990   \__physicx_matrix_set_r_c_ckall:nnn ,
991 check .default:n = all ,
992 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
993 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
994 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
995 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
996 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
997 diag+ .code:n =
998   \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
999 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
1000 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
1001 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
1002 item .clist_set:N = \l__physicx_matrix_item_clist ,
1003 item+ .code:n =
1004   \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
1005 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
1006 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
1007 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
1008 check-range .choice: ,
1009 check-range / true .code:n = \physicx_parse_range_check: ,
1010 check-range / false .code:n = \physicx_parse_range_noccheck: ,
1011 check-range .default:n = true ,
1012 begin .tl_set:N = \__physicx_matrix_begin:w ,

```

```

1013 end .tl_set:N = \__physicx_matrix_end: ,
1014 args .code:n =
1015 \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
1016 args* .tl_set:N = \l__physicx_matrix_args_tl ,
1017 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
1018 after-begin+ .code:n =
1019 { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
1020 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
1021 after-end+ .code:n =
1022 { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
1023 sepdim .dim_set:N = \l__physicx_matrix_sepdim ,
1024 type .multichoice: ,
1025 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
1026 saveto* .code:n =
1027 \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
1028 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
1029 transpose .default:n = true ,
1030 ' .meta:n = { transpose = true } ,
1031 T .meta:n = { transpose = true } ,
1032 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
1033 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
1034 enhanced .default:n = true ,
1035 !enhanced .code:n =
1036 \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
1037 cr .tl_set:N = \physicx@cr ,
1038 align .tl_set:N = \physicx@align ,
1039 sep .tl_set:N = \physicx@sep ,
1040 adi-order .choice: ,
1041 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
1042 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
1043 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
1044 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
1045 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
1046 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
1047 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
1048 beginning+ .code:n =
1049 \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
1050 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
1051 ending+ .code:n =
1052 \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
1053
1054 settype .code:n = \setmatritxttype #1 ,
1055
1056 unknown .code:n =
1057 \physicx_search_also:nnF
1058 {
1059     physicx/matrix/type ,
1060     physicx/matrix/expand ,
1061     physicx/matrix/element-code* ,
1062 }
1063 {#1}
1064 {
1065     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
1066     {

```

```

1067         \keys_set:nx { physicx/matrix }
1068         { MaxMatrixCols = \l_keys_key_str }
1069     }
1070     {
1071         \msg_error:nxxx { physicx } { unknown-key }
1072         \l_keys_path_str { physicx/matrix }
1073     }
1074 } ,
1075 }

```

`\physicx_matrix_new_type:nnn`
`\physicx_matrix_new_type:nn`
`\setmatrixtype`

```

1076 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
1077 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
1078 \cs_new:Npn \physicx_matrix_new_type:nn
1079 { \physicx_new_type:nnn { matrix } }
1080 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
1081 {
1082     \IfBooleanTF {#1}
1083     { \physicx_matrix_new_type:nn {#2} }
1084     { \physicx_matrix_new_type:nnn {#2} }
1085 }

```

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

A few types.

```

1086 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
1087 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
1088 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
1089 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
1090 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
1091 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
1092 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
1093 \physicx_mathtools:T
1094 {
1095     \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
1096     \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
1097     \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
1098     \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
1099     \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
1100     \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
1101     \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
1102     \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
1103     \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
1104     \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
1105     \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
1106     \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
1107     \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
1108     \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
1109     \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
1110     \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
1111     \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
1112 }

```

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

```

1113 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
1114 { \clist_set:cn { physicx@ #1 data@ #2 } }
1115 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
1116 {
1117   \clist_clear:c { l__physicx_matrix_ #1 _clist }
1118   \__physicx_matrix_add_data:nn {#1} {#2}
1119 }
1120 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
1121 {
1122   \clist_map_inline:nn {#2}
1123   {
1124     \clist_concat:ccc
1125     { l__physicx_matrix_ #1 _clist }
1126     { l__physicx_matrix_ #1 _clist }
1127     { physicx@ #1 data@ #2 }
1128   }
1129 }

```

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

Initial settings.

```

1130 \keys_set:nn { physicx/matrix }
1131 {
1132   type = m ,
1133   saveto = ? ,
1134 }

```

\qxmatri

```

1135 %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
1136   xmatmmn-in-the-physics-package, but changed some
1137 % #1 = boolean, saveto matrix
1138 % #2 = star, infinite
1139 % #3 = options
1140 % #4 = letter for the entries
1141 % #5 = number of rows
1142 % #6 = number of explicit rows, default = 3
1143 % #7 = number of columns
1144 % #8 = number of explicit columns, default = 3
1145 \DeclareDocumentCommand \qxmatri { t= s 0{type=p} m m 0{3} m 0{3} }
1146 {
1147   \group_begin:
1148   \IfBooleanTF { #2 }
1149   { \bool_set_true:N \l__physicx_matrix_infinite_bool }
1150   { \bool_set_false:N \l__physicx_matrix_infinite_bool }
1151   \int_set:Nn \l__physicx_matrix_rows_int {#6}
1152   \int_set:Nn \l__physicx_matrix_cols_int {#8}
1153   \IfBooleanTF {#1}
1154   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1155   { \keys_set:nn { physicx/matrix } {#3} }
1156   \physicx_qxmatri:nnn {#4} {#5} {#7}
1157   \__physicx_matrix_save_or_print:
1158   \group_end:
1159 }
1160 \cs_new_protected:Nn \physicx_qxmatri:nnn
1161 {

```

```

1161 \bool_if:NTF \l__physicx_matrix_expand_element_bool
1162 {
1163     \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1164     \__physicx_matrix_appto_body_e:nnn
1165 }
1166 {
1167     \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1168     \__physicx_matrix_appto_body_ne:nnn
1169 }
1170 % clear the variable containing the body of the matrix
1171 \tl_clear:N \l__physicx_matrix_body_tl
1172 % set the tentative number of explicit rows
1173 \physicx_if_num:NTF { #2 }
1174 {
1175     % number of rows is an integer
1176     \int_compare:NTF { #2 <= \l__physicx_matrix_rows_int }
1177     {
1178         % if #2 <= rows, we don't want a row of dots
1179         \bool_set_false:N \l__physicx_matrix_dotrow_bool
1180         \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1181     }
1182     {
1183         % we want a row of dots
1184         \bool_set_true:N \l__physicx_matrix_dotrow_bool
1185     }
1186 }
1187 % set the tentative number of explicit columns
1188 \physicx_if_num:NTF { #3 }
1189 {
1190     % number of cols is an integer
1191     \int_compare:NTF { #3 <= \l__physicx_matrix_cols_int }
1192     {
1193         % if #3 <= cols, we don't want a column of dots
1194         \bool_set_false:N \l__physicx_matrix_dotcol_bool
1195         \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1196     }
1197     {
1198         % we want a column of dots
1199         \bool_set_true:N \l__physicx_matrix_dotcol_bool
1200     }
1201 }
1202 % number of columns is symbolic, we want a column of dots
1203 \bool_set_true:N \l__physicx_matrix_dotcol_bool
1204 }
1205 % loop through the rows
1206 \int_step_inline:nn { \l__physicx_matrix_rows_int }
1207 {
1208     % add the first entry in the row
1209     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1210     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1211     % add the further entries in the explicit columns
1212     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1213     {
1214         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1215         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1216         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1217     }
1218 }

```



```

1215 % if we have a column of dots, add \cdots and the last entry
1216 \bool_if:NT \l__physicx_matrix_dotcol_bool
1217 {
1218     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1219     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1220     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1221 }
1222 % infinite matrix, add \cdots
1223 \bool_if:NT \l__physicx_matrix_infinite_bool
1224 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1225 \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
1226 \scan_stop:
1227 \else:
1228 % finish up the row
1229 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1230 \fi:
1231 }
1232 % finish up the rows
1233 \bool_if:NT \l__physicx_matrix_dotrow_bool
1234 {
1235 % finish up the row
1236 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1237 % if we have a row of dots, fill it in
1238 \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1239 \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1240 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1241 \bool_if:NT \l__physicx_matrix_dotcol_bool
1242 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1243 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1244 % fill the last row
1245 %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1246 \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1247 \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1248 {
1249     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1250     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1251     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1252 }
1253 \bool_if:NT \l__physicx_matrix_dotcol_bool
1254 {
1255     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1256     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1257     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1258 }
1259 % if the matrix is infinite, add a further column with \cdots
1260 \bool_if:NT \l__physicx_matrix_infinite_bool
1261 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1262 }
1263 % if the matrix is infinite, add a final row
1264 \bool_if:NT \l__physicx_matrix_infinite_bool
1265 {
1266 % finish up the row
1267 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1268 \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }

```

```

1269     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1270     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1271     \bool_if:NT \l__physicx_matrix_dotcol_bool
1272     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1273     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1274     % update cols
1275     \bool_if:NTF \l__physicx_matrix_dotcol_bool
1276     { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1277     { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1278   }
1279 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1280 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1281 {
1282   \keyval_parse:nnn
1283   \__physicx_matrix_diag_parse_aux:n
1284   \__physicx_matrix_diag_parse_aux:nn
1285   {#1}
1286 }
1287 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1288 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1289 {
1290   \str_case:e:nnF {#1}
1291   {
1292     { auto-update }
1293     {
1294       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1295       \__physicx_matrix_calc:nn
1296     }
1297     { noauto-update }
1298     {
1299       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1300     }
1301     { true }
1302     {
1303       \bool_set_true:N \l__physicx_matrix_diag_bool
1304       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1305       \__physicx_diagonalmatrix_set_diag:
1306     }
1307     { false }
1308     {
1309       \bool_set_false:N \l__physicx_matrix_diag_bool
1310       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1311       \__physicx_diagonalmatrix_no_diag:
1312     }
1313   }
1314   { \msg_error:nnn { physicx } { diag-key } {#1} }
1315 }
1316 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1317 {
1318   \tl_set:Nn \l__physicx_tmpdiag_tl {#2}

```

```

1319 \tl_set:Nx \l__physicx_tmpdiag_tl
1320 { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1321 \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
1322 \tl_if_head_eq_charcode:nNTF {#1} '
1323 {
1324   \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1325   { \tl_tail:n {#1} }
1326 }
1327 { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1328 }
1329 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1330 {
1331   \int_zero:N \l__physicx_matrix_cols_int
1332   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1333   {
1334     \int_incr:N \l__physicx_matrix_cols_int
1335     \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1336   }
1337   \int_set_eq:NN \l__physicx_matrix_rows_int
1338   \l__physicx_matrix_cols_int
1339 }
1340 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1341 {
1342   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1343   { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1344   \__physicx_matrix_diag_calc:nn
1345   { \seq_count:N \l__physicx_tmpdiag_seq }
1346   { \seq_count:N \l__physicx_tmpdiag_seq }
1347 }
1348 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1349 \__physicx_diagonalmatrix_no_diag:
1350 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1351 {
1352   \if_int_compare:w #1 = 0 \exp_stop_f:
1353   \__physicx_diagonalmatrix_diag_main:
1354   \else:
1355     \if_int_compare:w #1 > 0 \exp_stop_f:
1356     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1357     {
1358       \physicx_matrix_set_r_c:nnn
1359       {##1} { \int_eval:n { ##1 + #1 } } {##2}
1360     }
1361     \__physicx_matrix_diag_calc:nn
1362     { \seq_count:N \l__physicx_tmpdiag_seq }
1363     { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1364   \else:
1365     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1366     {
1367       \physicx_matrix_set_r_c:nnn
1368       { \int_eval:n { ##1 - #1 } } {##1} {##2}
1369     }
1370     \__physicx_matrix_diag_calc:nn
1371     { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1372     { \seq_count:N \l__physicx_tmpdiag_seq }

```

```

1373     \fi:
1374     \fi:
1375   }
1376   \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1377   {
1378     \if_int_compare:w #1 = 0 \exp_stop_f:
1379       \__physicx_matrix_diag_calc:nn
1380       { \seq_count:N \l__physicx_tmpdiag_seq }
1381       { \seq_count:N \l__physicx_tmpdiag_seq }
1382       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1383       {
1384         \physicx_matrix_set_r_c:nnn
1385         {##1}
1386         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1387         {##2}
1388       }
1389     \else:
1390       \if_int_compare:w #1 > 0 \exp_stop_f:
1391         \__physicx_matrix_diag_calc:nn
1392         { \seq_count:N \l__physicx_tmpdiag_seq }
1393         { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1394         \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1395         {
1396           \physicx_matrix_set_r_c:nnn
1397           {##1}
1398           { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1399           {##2}
1400         }
1401       \else:
1402         \__physicx_matrix_diag_calc:nn
1403         { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1404         { \seq_count:N \l__physicx_tmpdiag_seq }
1405         \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1406         {
1407           \physicx_matrix_set_r_c:nnn
1408           { \int_eval:n { ##1 - #1 } }
1409           { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1410           {##2}
1411         }
1412     \fi:
1413     \fi:
1414   }
1415   \cs_new:Npn \__physicx_matrix_diag_calc:nn
1416   { \__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
1417 \cs_new:Npn \physicx_matrix_item_parse:n #1
1418 {
1419   \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1420   \keyval_parse:NNn
1421   \__physicx_matrix_item_parse_aux:n
1422   \__physicx_matrix_item_parse_aux:nn

```

```

1423     {#1}
1424   }
1425   \cs_generate_variant:Nn \physics_matrix_item_parse:n { o }
1426   \cs_new:Npn \__physics_matrix_item_parse_aux:n #1 { }
1427   \cs_new:Npn \__physics_matrix_item_parse_aux:nn #1#2
1428   {
1429     \tl_set:Nn \l__physics_tmpitem_tl {#2}
1430     \tl_set:Nx \l__physics_tmpitem_tl
1431     { \__physics_expand:w \l__physics_tmpitem_tl }
1432     \physics_parse_range:neN \l__physics_matrix_rows_int
1433     { \use_i:nn #1 } \l__physics_tmp_rownum_seq
1434     \physics_parse_range:neN \l__physics_matrix_cols_int
1435     { \use_ii:nn #1 } \l__physics_tmp_colnum_seq
1436     \exp_args:No \tl_if_eq:nnTF
1437     { \l__physics_tmpitem_tl } { \PHYSICXIGNORE }
1438     {
1439       \seq_map_inline:Nn \l__physics_tmp_rownum_seq
1440       {
1441         \seq_map_inline:Nn \l__physics_tmp_colnum_seq
1442         {
1443           \clist_put_right:Nn \l__physics_item_ignore_clist { [##1][####1] }
1444         }
1445       }
1446     }
1447     {
1448       \seq_map_inline:Nn \l__physics_tmp_rownum_seq
1449       {
1450         \seq_map_inline:Nn \l__physics_tmp_colnum_seq
1451         {
1452           \clist_if_in:NnF \l__physics_item_ignore_clist { [##1][####1] }
1453           {
1454             \exp_args:Nnno \physics_matrix_set_r_c:nnn
1455             {##1} {####1} { \l__physics_tmpitem_tl }
1456           }
1457         }
1458       }
1459     }
1460   }

```

(End definition for \physics_matrix_item_parse:n. This function is documented on page ??.)

\physics_matrix_array_parse:n Parse ‘array...’ keys.

```

\physics_matrix_array_parse:o
1461   \cs_new:Npn \physics_matrix_array_parse:n #1
1462   {
1463     \tl_set:Nn \l__physics_tmparr_tl {#1}
1464     \tl_set:Nx \l__physics_tmparr_tl
1465     { \__physics_expand:w \l__physics_tmparr_tl }
1466     \seq_set_split:NVV \l__physics_matrix_tmparr_r_sep \physicsx@cr \l__physics_tmparr_tl
1467     \__physics_matrix_autocalc:nn
1468     { \seq_count:N \l__physics_matrix_tmparr_r_sep }
1469     { 0 }
1470     \seq_map_indexed_inline:Nn \l__physics_matrix_tmparr_r_sep
1471     {
1472       \seq_set_split:Non \l__physics_matrix_tmparr_c_sep { \physicsx@align } {##2}

```

```

1473     \__physicx_matrix_autocalc:nn
1474     { 0 }
1475     { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1476     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1477     {
1478         \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1479     }
1480 }
1481 }
1482 \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.

```

1483 \cs_new:Npn \physicx_matrix_array_parse_main:
1484 {
1485     \int_step_inline:nn \l__physicx_matrix_rows_int
1486     {
1487         \int_step_inline:nn \l__physicx_matrix_cols_int
1488         {
1489             \exp_args:Nno \physicx_matrix_set_r_c:nnn
1490             {##1} {####1} \l__physicx_matrix_main_tl
1491         }
1492     }
1493 }

```

(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).

```

1494 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1495 {
1496     \physicx_if_num:nTF {#1}
1497     { \prg_return_true: }
1498     {
1499         \bool_case_true:nTF
1500         {
1501             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1502             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1503             {
1504                 \bool_lazy_and_p:nn
1505                 { \cs_if_exist_p:N \inteval }
1506                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1507             } { }
1508             {
1509                 \bool_lazy_and_p:nn
1510                 { \cs_if_exist_p:N \fpeval }
1511                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1512             } { }
1513         }
1514         { \prg_return_true: }
1515         { \prg_return_false: }
1516     }
1517 }

```

(End definition for `_physicx_if_can_num:n`.)

`\diagonalmatrix` Define `\diagonalmatrix`.

```

1518 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{} m }
1519 {
1520   \group_begin:
1521   \IfBooleanTF {#1}
1522     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1523     { \keys_set:nn { physicx/matrix } { #3 } }
1524   \physicx_construct:nnn { }
1525   {
1526     \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1527     \tl_if_empty:nF {#4}
1528     {
1529       \__physicx_if_keyval:nTF {#4}
1530       { \physicx_matrix_diag_parse:n { true, #4 } }
1531       { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1532     }
1533   }
1534   { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1535   \bool_lazy_or:nnTF
1536     { \bool_if_p:n {#2} }
1537     { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1538   {
1539     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1540     {
1541       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1542       \__physicx_matrix_appto_body_e:off
1543     }
1544     {
1545       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1546       \__physicx_matrix_appto_body_ne:off
1547     }
1548     \use_i_ii:nnn
1549   }
1550   { \use_i:nn }
1551   \__physicx_matrix_transpose:N
1552   \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1553   \__physicx_diagonalmatrix_generate_body:NNN
1554   \__physicx_matrix_save_or_print:
1555   \group_end:
1556 }
1557 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1558 {
1559   \__physicx_matrix_generate_body:NNNN #1#2#3
1560   \__physicx_diagonalmatrix_enhanced:nnn
1561 }
1562 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1563 {
1564   \int_step_inline:nn { #1 - 1 }
1565   {
1566     \int_step_inline:nn { #2 - 1 }
1567     {
1568       \tl_put_right:Nx \l__physicx_matrix_body_tl

```

```

1569         {
1570             \exp_after:wN
1571             \physicx_matrix_use_r_c:nn
1572             #3 {{##1}} {{####1}} &
1573         }
1574     }
1575     \tl_put_right:Nx \l__physicx_matrix_body_tl
1576     {
1577         \exp_after:wN
1578         \physicx_matrix_use_r_c:nn
1579         #3 {{##1}} {{ \int_use:N #2 }} \__physicx_matrix_sep:
1580     }
1581 }
1582 \int_step_inline:nn { #2 - 1 }
1583 {
1584     \tl_put_right:Nx \l__physicx_matrix_body_tl
1585     {
1586         \exp_after:wN
1587         \physicx_matrix_use_r_c:nn
1588         #3 {{ \int_use:N #1 }} {{##1}} &
1589     }
1590 }
1591 \tl_put_right:Nx \l__physicx_matrix_body_tl
1592 {
1593     \exp_after:wN
1594     \physicx_matrix_use_r_c:nn
1595     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1596 }
1597 }

```

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

__physicx_declare_init:

```

1598 \cs_new:Npn \__physicx_matrix_enhanced_init:
1599 {
1600     \seq_if_empty:NF \l__physicx_row_list_seq
1601     {
1602         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1603         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1604             { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1605     }
1606     \seq_if_empty:NF \l__physicx_col_list_seq
1607     {
1608         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1609         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1610             { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1611     }
1612 }

```

(End definition for __physicx_declare_init:.)

\commamatrix Define \commamatrix.

```

1613 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1614 {
1615     \group_begin:

```



```

1616 \keys_set:nn { physicx/matrix } {#3}
1617 \tl_if_empty:nF {#4}
1618 { \keys_set:nn { physicx/matrix } { array = {#4} } }
1619 \IfBooleanT {#1}
1620 { \keys_set:nn { physicx/matrix } { saveto = \physicxtmp } }
1621 \tl_set:Nx \l__physicx_matrix_array_tl
1622 { \__physicx_expand:w \l__physicx_matrix_array_tl }
1623 \bool_lazy_or:nnTF
1624 { \bool_if_p:n {#2} }
1625 { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1626 { \__physicx_commamatrix_enhanced: }
1627 {
1628 \tl_replace_all:Nox \l__physicx_matrix_array_tl
1629 { \physicx@cr } { \__physicx_matrix_sep: }
1630 \tl_replace_all:Non \l__physicx_matrix_array_tl
1631 { \physicx@align } { & }
1632 \tl_set_eq:NN \l__physicx_matrix_body_tl
1633 \l__physicx_matrix_array_tl
1634 }
1635 \__physicx_matrix_save_or_print:
1636 \group_end:
1637 }
1638 \cs_new_nopar:Npn \__physicx_matrix_save_or_print:
1639 {
1640 \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1641 {
1642 \exp_after:wN \tl_gset_eq:NN
1643 \l__physicx_matrix_save_tl
1644 \l__physicx_matrix_body_tl
1645 }
1646 {
1647 \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1648 \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1649 \fi:
1650 \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matri
1651 \l__physicx_matrix_body_tl
1652 \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1653 }
1654 }
1655 \cs_new:Npn \__physicx_commamatrix_enhanced:
1656 {
1657 \tl_clear:N \l__physicx_matrix_body_tl
1658 \int_zero:N \l__physicx_tmpa_int
1659 \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1660 \l__physicx_matrix_array_tl
1661 \int_set:Nn \l__physicx_matrix_rows_int
1662 { \seq_count:N \l__physicx_tmp_seq }
1663 \__physicx_matrix_enhanced_init:
1664 \bool_if:NTF \l__physicx_matrix_expand_element_bool
1665 {
1666 \seq_map_tokens:Nn \l__physicx_tmp_seq
1667 {
1668 \int_incr:N \l__physicx_tmpa_int
1669 \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn

```

```

1670         \l__physicx_tmpa_int \l__physicx_commamatrix_enhanced_aux_e:nnn
1671     }
1672 }
1673 {
1674     \seq_map_tokens:Nn \l__physicx_tmp_seq
1675     {
1676         \int_incr:N \l__physicx_tmpa_int
1677         \exp_args:NV \l__physicx_commamatrix_enhanced_aux:nNn
1678         \l__physicx_tmpa_int \l__physicx_commamatrix_enhanced_aux_ne:nnn
1679     }
1680 }
1681 }
1682 \cs_new:Npn \l__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1683 {
1684     \seq_set_split:Non \l__physicx_tmp_col_seq
1685     { \physicx@align } {#3}
1686     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1687     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1688     { #2 {##2} {#1} {##1} }
1689     \tl_put_right:Nx \l__physicx_matrix_body_tl
1690     {
1691         \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1692         \if_int_compare:w \l__physicx_matrix_rows_int = #1
1693             \scan_stop:
1694         \else:
1695             \l__physicx_matrix_sep:
1696         \fi:
1697     }
1698 }
1699 \cs_new:Npn \l__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1700 {
1701     \seq_put_right:Nx \l__physicx_tmp_coled_seq
1702     {
1703         \text_expand:n % \text_expand:n do the magic thing, but slower
1704         {
1705             \physicx@matricelement { #1 }
1706             { \l__physicx_matrix_row_iterate:n {#2} }
1707             { \l__physicx_matrix_col_iterate:n {#3} }
1708         }
1709     }
1710 }
1711 \cs_new:Npn \l__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1712 {
1713     \seq_put_right:No \l__physicx_tmp_coled_seq
1714     {
1715         \physicx@matricelement {#1}
1716         { \l__physicx_matrix_row_iterate:n {#2} }
1717         { \l__physicx_matrix_col_iterate:n {#3} }
1718     }
1719 }

```

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

\generalmatrix Define \generalmatrix.

```

1720 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1721 {
1722   \IfBooleanTF {#2}
1723   {
1724     \group_begin:
1725     \IfBooleanTF {#1}
1726     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1727     { \keys_set:nn { physicx/matrix } {#4} }
1728     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1729     \physicx_construct:nnn
1730     {
1731       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1732       {
1733         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1734       }
1735       { \physicx_matrix_array_parse_main: }
1736     }
1737     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1738     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1739     \__physicx_generalmatrix:
1740     \__physicx_matrix_save_or_print:
1741     \group_end:
1742   }
1743   {
1744     \IfBooleanTF {#1}
1745     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1746     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1747     \qxmatrix = * [#4]
1748   }
1749 }
1750 \cs_new:Npn \__physicx_generalmatrix:
1751 {
1752   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1753   {
1754     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1755     \__physicx_matrix_appto_body_e:off
1756   }
1757   {
1758     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1759     \__physicx_matrix_appto_body_ne:off
1760   }
1761   \__physicx_matrix_transpose:N
1762   \__physicx_matrix_generate_body:NNNN
1763   \__physicx_generalmatrix_generate:nnn
1764 }

```

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

__physicx_matrix_generate_body:NNNN

```

1765 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1766 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1767 {
1768   \__physicx_matrix_enhanced_init:
1769   \int_step_inline:nn { #1 - 1 }

```

```

1770 {
1771   \int_step_inline:nn { #2 - 1 }
1772   {
1773     \tl_set:Nx \l__physicx_tmp_tl
1774     {
1775       \exp_after:wN
1776       \physicx_matrix_use_r_c:nn
1777       #3 {{##1}} {{####1}}
1778     }
1779     #4 \l__physicx_tmp_tl {##1} {####1}
1780     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1781   }
1782   \tl_set:Nx \l__physicx_tmp_tl
1783   {
1784     \exp_after:wN
1785     \physicx_matrix_use_r_c:nn
1786     #3 {{##1}} {{ \int_use:N #2 }}
1787   }
1788   #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1789   \tl_put_right:Nx \l__physicx_matrix_body_tl
1790   { \__physicx_matrix_sep: }
1791 }
1792 \int_step_inline:nn { #2 - 1 }
1793 {
1794   \tl_set:Nx \l__physicx_tmp_tl
1795   {
1796     \exp_after:wN
1797     \physicx_matrix_use_r_c:nn
1798     #3 {{ \int_use:N #1 }} {{##1}}
1799   }
1800   #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1801   \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1802 }
1803 \tl_set:Nx \l__physicx_tmp_tl
1804 {
1805   \exp_after:wN
1806   \physicx_matrix_use_r_c:nn
1807   #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1808 }
1809 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1810 }

```

(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
1811 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1812 {
1813   \tl_put_right:Nx \l__physicx_matrix_body_tl
1814   {
1815     \text_expand:n
1816     {
1817       \physicx@matricelement {#1}
1818       { \__physicx_matrix_row_iterate:n {#2} }
1819       { \__physicx_matrix_col_iterate:n {#3} }

```

```

1820     }
1821   }
1822 }
1823 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1824 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1825 {
1826   \tl_put_right:No \l__physicx_matrix_body_tl
1827   {
1828     \physicx@matricelement {#1}
1829     { \__physicx_matrix_row_iterate:n {#2} }
1830     { \__physicx_matrix_col_iterate:n {#3} }
1831   }
1832 }
1833 \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

```

1834 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1835 {
1836   \bool_if:NTF \l__physicx_matrix_transpose_bool
1837   {
1838     #1
1839     \l__physicx_matrix_cols_int
1840     \l__physicx_matrix_rows_int
1841     \use_ii_i:nn
1842   }
1843   {
1844     #1
1845     \l__physicx_matrix_rows_int
1846     \l__physicx_matrix_cols_int
1847     \use:nn
1848   }
1849 }

```

(End definition for __physicx_matrix_transpose:N.)

__physicx_matrix_sep:

```

1850 \cs_new:Npn \__physicx_matrix_sep:
1851 {
1852   \dim_compare:nNnTF \l__physicx_matrix_sep_dim = \c_zero_dim
1853   { \ \ } { \ \ [ \dim_use:N \l__physicx_matrix_sep_dim ] }
1854 }

```

(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.

```

1855 \cs_new:Npn \physicx_construct:nnn #1#2#3
1856 {
1857   \l__physicx_matrix_beginning_tl
1858   \__physicx_adi:nnn {#1} {#2} {#3}
1859   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1860   {
1861     \int_incr:N \l__physicx_matrix_cols_int

```

```

1862     \__physicx_matrix_last_aux_c:
1863     \int_incr:N \l__physicx_matrix_cols_int
1864   }
1865   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1866   {
1867     \int_incr:N \l__physicx_matrix_rows_int
1868     \__physicx_matrix_last_aux_r:
1869     \int_incr:N \l__physicx_matrix_rows_int
1870   }
1871   \bool_lazy_or:nnF
1872   { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1873   { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1874   {
1875     \physicx_matrix_set_r_c:nnn
1876     { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1877     { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1878     { \ddots }
1879   }
1880   \bool_if:NT \l__physicx_matrix_infinite_bool
1881   {
1882     \int_incr:N \l__physicx_matrix_rows_int
1883     \int_incr:N \l__physicx_matrix_cols_int
1884     \__physicx_matrix_last_aux_c:
1885     \__physicx_matrix_last_aux_r:
1886     \physicx_matrix_set_r_c:nnn
1887     { \int_use:N \l__physicx_matrix_rows_int }
1888     { \int_use:N \l__physicx_matrix_cols_int }
1889     { \ddots }
1890   }
1891   \l__physicx_matrix_ending_tl
1892 }
1893 \cs_new:Npn \__physicx_matrix_last_aux_c:
1894 {
1895   \int_step_inline:nn \l__physicx_matrix_rows_int
1896   {
1897     \physicx_matrix_set_r_c:nnn
1898     {##1} { \int_use:N \l__physicx_matrix_cols_int }
1899     { \cdots }
1900   }
1901 }
1902 \cs_new:Npn \__physicx_matrix_last_aux_r:
1903 {
1904   \int_step_inline:nn \l__physicx_matrix_cols_int
1905   {
1906     \physicx_matrix_set_r_c:nnn
1907     { \int_use:N \l__physicx_matrix_rows_int } {##1}
1908     { \vdots }
1909   }
1910 }

```

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

1.3.3 Define new matrix command

```

1911 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
1912 {
1913   \NewDocumentCommand #2 { t+ m o o m m }
1914   {
1915     \IfBooleanTF {##1}
1916     {
1917       \IfNoValueTF {##3}
1918       { \newcommand ##2 { #1 + [##5] {##6} } }
1919       {
1920         \IfNoValueTF {##4}
1921         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1922         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1923       }
1924     }
1925     {
1926       \IfNoValueTF {##3}
1927       { \newcommand ##2 { #1 [##5] {##6} } }
1928       {
1929         \IfNoValueTF {##4}
1930         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1931         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1932       }
1933     }
1934   }
1935   \NewDocumentCommand #3 { t+ m m m m }
1936   {
1937     \IfBooleanTF {##1}
1938     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1939     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1940   }
1941 }
1942 \__physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1943 \__physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1944 \NewDocumentCommand \newgeneralmatrix { t+ m o o m m }
1945 {
1946   \IfBooleanTF {#1}
1947   {
1948     \IfNoValueTF {#3}
1949     { \newcommand #2 { \generalmatrix + {#5} } }
1950     {
1951       \IfNoValueTF {#4}
1952       { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1953       { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1954     }
1955   }
1956   {
1957     \IfNoValueTF {#3}
1958     { \newcommand #2 { \generalmatrix {#5} } }
1959     {
1960       \IfNoValueTF {#4}
1961       { \newcommand #2 [#3] { \generalmatrix {#5} } }

```

```

1962         { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1963     }
1964 }
1965 }
1966 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1967 {
1968     \IfBooleanTF {#1}
1969     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1970     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1971 }

```

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

```

1972 \</package>

```

Index

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

| Symbols | | |
|----------------------------------|-------|--|
| <code>\+</code> | | 177 |
| <code>\-</code> | | 177 |
| <code>\.</code> | | 177 |
| <code>\ </code> | | 57 , 1853 |
| <code>\{</code> | | 440 , 678 , 697 , 715 , 756 , 781 |
| <code>\}</code> | | 440 , 678 , 699 , 715 , 756 , 781 |
| <code>\sqcup</code> | | 177 |
| A | | |
| <code>\A</code> | | 177 |
| <code>\abs</code> | | 749 |
| <code>\absolutevalue</code> | | 749 |
| <code>\acommm</code> | | 749 |
| <code>\aftergroup</code> | | 227 |
| <code>\anticommutator</code> | | 749 |
| <code>\AtBeginDocument</code> | | 272 , 295 , 308 , 319 , 336 |
| B | | |
| <code>\begin</code> | | 445 , 446 , 447 , 448 , 449 , 450 , 451 , 454 , 455 , 456 , 457 , 458 , 459 , 460 , 461 , 462 , 463 , 464 , 465 , 466 , 467 , 468 , 469 , 470 , 698 , 701 , 705 , 708 , 709 , 710 , 715 , 716 , 720 , 723 , 724 , 725 , 1086 , 1087 , 1088 , 1089 , 1090 , 1091 , 1092 , 1095 , 1096 , 1097 , 1098 , 1099 , 1100 , 1101 , 1102 , 1103 , 1104 , 1105 , 1106 , 1107 , 1108 , 1109 , 1110 , 1111 |
| <code>\bgroup</code> | | 226 , 228 |
| <code>\Big</code> | | 731 |
| <code>\big</code> | | 731 |
| <code>\Bigg</code> | | 731 |
| <code>\bigg</code> | | 731 |
| <code>\Biggl</code> | | 738 |
| <code>\biggl</code> | | 737 |
| <code>\Biggr</code> | | 738 |
| <code>\biggr</code> | | 737 |
| <code>\Bigl</code> | | 736 |
| <code>\bigl</code> | | 735 |
| <code>\Bigr</code> | | 736 |
| <code>\bigr</code> | | 735 |
| <code>\boldsymbol</code> | | 216 |
| bool commands: | | |
| <code>\bool_case_false:n</code> | | 648 , 649 , 656 |
| <code>\bool_case_true:nTF</code> | | 733 , 1499 |
| <code>\bool_if:N</code> | | 16 , 21 , 26 , 85 , 205 , 270 , 311 , 317 , 330 , 662 , 664 , 668 , 1161 , 1216 , 1223 , 1233 , 1241 , 1253 , 1260 , 1264 , 1271 , 1275 , 1539 , 1664 , 1752 , 1836 , 1880 |
| <code>\bool_if_p:N</code> | | 1537 , 1625 |
| <code>\bool_if_p:n</code> | | 735 , 736 , 737 , 738 , 1536 , 1624 |
| <code>\bool_lazy_and:p:nn</code> | | 1504 , 1509 |
| <code>\bool_lazy_or:nnTF</code> | | 31 , 137 , 152 , 493 , 498 , 1535 , 1623 , 1871 |
| <code>\bool_new:N</code> | | 9 , 10 , 11 , 12 , 13 , 51 , 66 , 564 , 566 , 840 , 841 , 842 , 850 , 864 , 872 , 873 |
| <code>\bool_set:Nn</code> | | 575 , 576 , 1728 |
| <code>\bool_set_false:N</code> | | 95 , 193 , 282 , 1149 , 1177 , 1192 , 1309 |

| | | | |
|--|---|--|---|
| <code>\bool_set_inverse:N</code> | 910, 1036 | <code>\cs_new_protected:Nn</code> | 1159 |
| <code>\bool_set_true:N</code> | 140, | <code>\cs_new_protected:Npn</code> | 67, |
| 155, 165, 200, 268, 281, 1148, 1181, | | 72, 77, 90, 93, 210, 338, 342, 348, 354 | |
| 1185, 1196, 1200, 1303, 1602, 1608 | | <code>\cs_new_protected_nopar:Npn</code> | |
| <code>\c_false_bool</code> | 693, 713 | | 230, 232, 1113, 1115, 1120 |
| <code>\c_true_bool</code> | 676, 686, 693, 713 | <code>\cs_set:Nn</code> | |
| <code>\Bqty</code> | 749 | | 1041, 1042, 1043, 1044, 1045, 1046 |
| <code>\bqty</code> | 749 | <code>\cs_set:Npn</code> | |
| | | | 114, 116, 124, 126, 294, 525, |
| | | | 531, 535, 752, 762, 763, 779, 784, |
| | | | 785, 786, 916, 925, 934, 943, 952, 961 |
| | | <code>\cs_set:Npx</code> | 968 |
| | | <code>\cs_set_eq:NN</code> | 69, 70, 74, 75, 216, |
| | | | 220, 224, 274, 275, 276, 309, 310, |
| | | | 311, 322, 323, 324, 333, 832, 885, |
| | | | 887, 889, 892, 898, 900, 914, 923, |
| | | | 932, 941, 950, 959, 977, 980, 983, |
| | | | 986, 989, 1163, 1167, 1294, 1299, |
| | | | 1304, 1310, 1541, 1545, 1754, 1758 |
| | | <code>\cs_set_nopar:Npn</code> | 1603, 1609 |
| | | <code>\cs_set_protected:Npn</code> | |
| | | | 640, 653, 672, 747 |
| | | <code>\cs_to_str:N</code> | |
| | | | 340, 342, 540, 543, 548, 549, 585 |
| C | | | |
| <code>\cdots</code> | 936, 1215, 1218, 1219, 1222, | | |
| 1224, 1255, 1256, 1259, 1261, 1899 | | | |
| clist commands: | | | |
| <code>\clist_clear:N</code> | 1117 | | |
| <code>\clist_concat:NNN</code> | 1124 | | |
| <code>\clist_if_empty:nTF</code> | 242 | | |
| <code>\clist_if_in:NnTF</code> | 1452 | | |
| <code>\clist_map_break:n</code> | 186, 198 | | |
| <code>\clist_map_inline:nn</code> 82, 182, 194, 1122 | | | |
| <code>\clist_new:N</code> | 50, 848, 849 | | |
| <code>\clist_put_right:Nn</code> | 998, 1004, 1443 | | |
| <code>\clist_set:Nn</code> | 1114 | | |
| <code>\clist_set_eq:NN</code> | 1419 | | |
| <code>\c_empty_clist</code> | 1419 | | |
| <code>\comm</code> | 779 | | |
| <code>\commamatrix</code> | 32, 1613, 1943 | | |
| <code>\commutator</code> | 749 | | |
| <code>\crossproduct</code> | 304 | | |
| cs commands: | | | |
| <code>\cs:w</code> | 32, 33, 1027 | | |
| <code>\cs_end:</code> | 32, 33, 803, 1027 | | |
| <code>\cs_generate_variant:Nn</code> | | | |
| | 3, 4, 5, 6, 89, 92, 208, | | |
| | 520, 659, 1287, 1425, 1482, 1823, 1833 | | |
| <code>\cs_gset_eq:NN</code> | 40, 41, 42 | | |
| <code>\cs_if_exist:NTF</code> | 411 | | |
| <code>\cs_if_exist_p:N</code> | 1505, 1510 | | |
| <code>\cs_if_exist_use:NTF</code> | 37 | | |
| <code>\cs_if_free:NTF</code> | 340 | | |
| <code>\cs_new:Npn</code> | 7, 110, | | |
| 135, 143, 146, 162, 173, 180, 214, | | | |
| 218, 222, 226, 227, 228, 229, 240, | | | |
| 396, 405, 478, 490, 569, 579, 591, | | | |
| 628, 635, 660, 729, 875, 1076, 1078, | | | |
| 1280, 1288, 1316, 1329, 1340, 1350, | | | |
| 1376, 1415, 1417, 1426, 1427, 1461, | | | |
| 1483, 1557, 1562, 1598, 1655, 1682, | | | |
| 1699, 1711, 1750, 1766, 1811, 1824, | | | |
| 1834, 1850, 1855, 1893, 1902, 1911 | | | |
| <code>\cs_new_eq:NN</code> | | | |
| | 145, 172, 834, 854, 863, 1348 | | |
| <code>\cs_new_nopar:Npn</code> | | | |
| | 234, 793, 801, 810, 812, | | |
| | 817, 823, 856, 857, 860, 861, 862, 1638 | | |
| | | <code>\d</code> | 177 |
| | | <code>\ddots</code> | 936, 1242, 1273, 1878, 1889 |
| | | <code>\DeclareDocumentCommand</code> | 296, |
| | | | 298, 300, 303, 304, 305, 312, 313, |
| | | | 645, 655, 731, 1144, 1518, 1613, 1720 |
| | | <code>\diagonalmatrix</code> | 31, 1518, 1942 |
| | | <code>\diffd</code> | 311 |
| | | dim commands: | |
| | | <code>\dim_compare:nNnTF</code> | 1852 |
| | | <code>\dim_new:N</code> | 874 |
| | | <code>\dim_use:N</code> | 1853 |
| | | <code>\c_zero_dim</code> | 1852 |
| | | <code>\div</code> | 274, 309, 310, 324 |
| | | <code>\divergence</code> | 310 |
| | | <code>\divisionsymbol</code> | 274, 309, 324 |
| | | <code>\dotproduct</code> | 303 |
| E | | | |
| | | <code>\egroup</code> | 227, 229 |
| | | else commands: | |
| | | <code>\else:</code> | |
| | | | 805, 1227, 1354, 1364, 1389, 1401, 1694 |
| | | <code>\end</code> | 445, |
| | | | 446, 447, 448, 449, 450, 451, 454, |
| | | | 455, 456, 457, 458, 459, 460, 461, |
| | | | 462, 463, 464, 465, 466, 467, 468, |
| | | | 469, 470, 698, 701, 705, 708, 709, |
| | | | 710, 715, 716, 720, 723, 724, 725, |

1086, 1087, 1088, 1089, 1090, 1091,
 1092, 1095, 1096, 1097, 1098, 1099,
 1100, 1101, 1102, 1103, 1104, 1105,
 1106, 1107, 1108, 1109, 1110, 1111
 \eval 749
 \evaluated 674, 762
 exp commands:
 \exp_after:wN
 ... 1570, 1577, 1586, 1593, 1640,
 1642, 1650, 1775, 1784, 1796, 1805
 \exp_args:Nc 352, 539, 542
 \exp_args:Nf 238, 1324
 \exp_args:NNc 351
 \exp_args:NNno 1454, 1489
 \exp_args:NNnx 356
 \exp_args:Nno 184, 196
 \exp_args:NNx 545
 \exp_args:No
 ... 420, 503, 954, 963, 1065, 1436
 \exp_args:NV 1669, 1677
 \exp_end: 8
 \exp_not:N 8, 414, 547, 548, 549, 599, 968
 \exp_not:n
 ... 603, 604, 804, 806, 854, 885, 889
 \exp_stop_f:
 ... 594, 601, 1352, 1355, 1378, 1390

F
 \fbraces 312, 313
 fi commands:
 \fi: 605,
 620, 625, 626, 727, 788, 807, 1230,
 1373, 1374, 1412, 1413, 1649, 1696
 fp commands:
 \fp_compare:nNnTF 236
 \fp_eval:n 30, 238, 401, 1502
 \fpeval 30, 1510, 1511

G
 \generalmatrix .. 34, 1720, 1949, 1952,
 1953, 1958, 1961, 1962, 1969, 1970
 group commands:
 \group_begin:
 ... 480, 1146, 1520, 1615, 1724
 \group_end: 488, 1157, 1555, 1636, 1741

H
 \hat 301
 hook commands:
 \hook_gput_code:nnn 37, 332, 334

I
 if commands:
 \if_bool:N 609, 621, 674, 749
 \if_case:w 601
 \if_cs_exist:w 803
 \if_int_compare:w 594, 1225,
 1352, 1355, 1378, 1390, 1647, 1692
 \IfBooleanT 697, 699, 1619
 \IfBooleanTF .. 297, 299, 301, 361, 556,
 647, 704, 706, 719, 721, 741, 1082,
 1147, 1152, 1521, 1722, 1725, 1744,
 1745, 1746, 1915, 1937, 1946, 1968
 \IfNoValueTF
 .. 312, 313, 523, 529, 1917, 1920,
 1926, 1929, 1948, 1951, 1957, 1960
 \Im 276, 313, 323
 \imaginary 276, 313, 323
 int commands:
 \int_compare:nNnTF 148, 150, 164
 \int_compare:nTF 1175, 1190
 \int_compare_p:nNn . 138, 139, 153, 154
 \int_eval:n .. 30, 1359, 1368, 1386,
 1398, 1408, 1409, 1501, 1876, 1877
 \int_incr:N . 593, 1334, 1668, 1676,
 1861, 1863, 1867, 1869, 1882, 1883
 \int_max:nn 796, 798
 \int_new:N
 .. 52, 53, 62, 63, 64, 65, 568, 845, 846
 \int_set:Nn .. 80, 81, 103, 106, 574,
 795, 797, 1150, 1151, 1178, 1193, 1661
 \int_set_eq:NN
 ... 102, 105, 149, 151, 1337, 1648
 \int_step_inline:nn
 ... 1203, 1485, 1487, 1564, 1566,
 1582, 1769, 1771, 1792, 1895, 1904
 \int_step_inline:nnn
 ... 157, 167, 1209, 1247
 \int_use:N 1579,
 1588, 1595, 1786, 1788, 1798, 1800,
 1807, 1809, 1887, 1888, 1898, 1907
 \int_zero:N 1331, 1658
 \interval 30, 1505, 1506

K
 keys commands:
 \keys_define:nn
 ... 211, 245, 373, 375, 878, 880
 \keys_if_exist:nnTF 184, 196
 \l_keys_key_str 184, 187, 196,
 201, 407, 411, 414, 420, 422, 1065, 1068
 \l_keys_path_str 432, 1072
 \keys_set:nn 3, 187, 201,
 362, 363, 374, 413, 422, 472, 481,
 879, 1067, 1130, 1153, 1154, 1522,
 1523, 1616, 1618, 1620, 1726, 1727
 keyval commands:
 \keyval_parse:Nn 1420
 \keyval_parse:nnn 344, 1282

| L | | P | |
|----------------------------------|--|---|--|
| <code>\langle</code> | 444, 682, 709, 724 | <code>\PassOptionsToPackage</code> | 242 |
| <code>\lbrace</code> | 312, 313 | <code>\pb</code> | 749 |
| <code>\ldots</code> | 936 | peek commands: | |
| <code>\left</code> | 226, 474, 499, 697, 704, 709, 715, 716, 719, 723, 724, 725 | <code>\peek_charcode_ignore_spaces:N</code> | 642 |
| <code>\let</code> | 312, 313, 642 | physics commands: | |
| <code>\letdif</code> | 335 | <code>\physicsx_bf:</code> | 216, 220, 224, 297, 299, 301 |
| <code>\lgroup</code> | 704, 719 | <code>\physicsx_compat:</code> | 14 |
| <code>\lVert</code> | 443, 766 | <code>\physicsx_compat:TF</code> | 284, 767 |
| <code>\lvert</code> | 442 | <code>\physicsx_construct:nnn</code> | 1524, 1729, 1855, 1855 |
| M | | <code>\physicsx_declare_legacy_paren:NnnnNNn</code> | 728, 729, 748, 753, 754, 755, 756, 758, 765, 769, 773, 776, 780, 789, 791 |
| <code>\mathbf</code> | 297, 299, 301 | <code>\physicsx_declare_legacy_quantity:nnNn</code> | 562, 579, 673, 675, 685, 692, 712 |
| <code>\mathcal</code> | 212, 213 | <code>\physicsx_if_num:n</code> | 175 |
| <code>\mathclose</code> | 226, 228 | <code>\physicsx_if_num:nTF</code> | 398, 420, 1065, 1173, 1188, 1496 |
| <code>\mathopen</code> | 226, 228 | <code>\physicsx_left:</code> | 226, 231, 501, 643, 743 |
| <code>\matrixquantity</code> | 674 | <code>\physicsx_left:N</code> | 228, 511, 735, 736, 737, 738 |
| <code>\mqty</code> | 749 | <code>\physicsx_left:nN</code> | 230, 504 |
| msg commands: | | <code>\physicsx_mathtools:</code> | 24 |
| <code>\msg_error:nnn</code> | 1314 | <code>\physicsx_mathtools:TF</code> | 452, 1093 |
| <code>\msg_error:nnnn</code> | 431, 1071 | <code>\physicsx_matrix_array_parse:n</code> | 1461, 1461, 1482, 1733 |
| <code>\msg_new:nnn</code> | 60 | <code>\physicsx_matrix_array_parse_-main:</code> | 1483, 1483, 1735 |
| <code>\msg_new:nnnn</code> | 54 | <code>\physicsx_matrix_diag_parse:n</code> | 999, 1280, 1280, 1287, 1526, 1530, 1531, 1737 |
| N | | <code>\physicsx_matrix_item_parse:n</code> | 1005, 1417, 1417, 1425, 1534, 1738 |
| <code>\nabla</code> | 294, 333 | <code>\physicsx_matrix_new_type:nn</code> | 1076, 1078, 1083 |
| <code>\NewCommaMatrix</code> | 1911 | <code>\physicsx_matrix_new_type:nnn</code> | 1076, 1076, 1084 |
| <code>\newcommamatrix</code> | 1911 | <code>\physicsx_matrix_set_r_c:nnn</code> | 834, 977, 980, 983, 986, 989, 1335, 1343, 1358, 1367, 1384, 1396, 1407, 1454, 1478, 1489, 1875, 1886, 1897, 1906 |
| <code>\newcommand</code> | 526, 532, 536, 545, 1918, 1921, 1922, 1927, 1930, 1931, 1949, 1952, 1953, 1958, 1961, 1962 | <code>\physicsx_matrix_use_r_c:nn</code> | 801, 1571, 1578, 1587, 1594, 1776, 1785, 1797, 1806 |
| <code>\NewDiagonalMatrix</code> | 1911 | <code>\physicsx_new_type:nnn</code> | 210, 438, 1077, 1079 |
| <code>\newdiagonalmatrix</code> | 1911 | <code>\physicsx_option_or:nn</code> | 29 |
| <code>\newdif</code> | 335 | <code>\physicsx_option_or:nnTF</code> | 750, 760, 778, 782 |
| <code>\NewDocumentCommand</code> | 359, 437, 519, 521, 552, 554, 1080, 1913, 1935, 1938, 1939, 1944, 1966, 1969, 1970 | <code>\c_physicsx_Order_tl</code> | 213, 287, 770, 790 |
| <code>\NewGeneralMatrix</code> | 1911 | <code>\c_physicsx_order_tl</code> | 212, 286, 774, 792 |
| <code>\newgeneralmatrix</code> | 1911 | <code>\physicsx_parse_range:nnN</code> | 90, 92, 1432, 1434 |
| <code>\NewXQuantity</code> | 478 | | |
| <code>\newxquantity</code> | 478 | | |
| <code>\nobreak</code> | 688, 689, 690 | | |
| <code>\norm</code> | 749 | | |
| O | | | |
| <code>\OOrder</code> | 789 | | |
| <code>\oorder</code> | 749 | | |
| <code>\operatorname</code> | 312, 313 | | |
| or commands: | | | |
| <code>\or:</code> | 602, 603, 604 | | |
| <code>\order</code> | 749 | | |
| <code>\Ordersymbol</code> | 287 | | |
| <code>\ordersymbol</code> | 286 | | |

| | |
|--|--|
| <code>\physicx_parse_range:nnnN</code> | <code>__physicx_declare_init:</code> |
| 62, 77, 89, 91 | <code>__physicx_declare_init:nnn</code> 569, 581 |
| <code>\physicx_parse_range_check:</code> | <code>__physicx_declare_legacy_-</code> |
| 62, 67, 1009 | <code>quantity_aux:nnnn</code> 591, 632 |
| <code>\physicx_parse_range_nocheck:</code> ... | <code>__physicx_declare_legacy_-</code> |
| 62, 72, 1010 | <code>quantity_aux:NNnnn</code> .. 584, 635, 659 |
| <code>\physicx_qxmatrix:nnn</code> ... 1155, 1159 | <code>__physicx_declare_legacy_-</code> |
| <code>\physicx_right:</code> 227, 233, 501, 643, 743 | <code>quantity_aux:nw</code> 582, 628, 633 |
| <code>\physicx_right:N</code> | <code>__physicx_delsize:NNnn</code> 231, 233, 234 |
| 229, 513, 735, 736, 737, 738 | <code>__physicx_diagonalmatrix_diag_-</code> |
| <code>\physicx_right:nN</code> | <code>main:</code> 1304, 1310, 1348, 1353 |
| <code>\physicx_search_also:nn</code> | <code>__physicx_diagonalmatrix_-</code> |
| 180, 191, 208, 209 | <code>enhanced:nnn</code> 1541, 1545, 1560 |
| <code>\physicx_search_also:nnTF</code> . 425, 1057 | <code>__physicx_diagonalmatrix_-</code> |
| <code>\physicx_set_parse_range_-</code> | <code>generate_body:NNN</code> 1553, 1562 |
| <code>delimiter:n</code> 110, 134 | <code>__physicx_diagonalmatrix_-</code> |
| <code>\physicx_short:</code> 19 | <code>generate_enhanced_body:NNN</code> .. |
| <code>\physicx_unimath:</code> 44 | 1552, 1557 |
| <code>\physicx_unimath:TF</code> .. 40, 41, 42, 293 | <code>__physicx_diagonalmatrix_no_-</code> |
| <code>\physicx_use_amsymb_type:</code> . 214, 292 | <code>diag:</code> 1311, 1340, 1349 |
| <code>\physicx_use_uni_bf_type:</code> 222 | <code>__physicx_diagonalmatrix_set_-</code> |
| <code>\physicx_use_uni_bfit_type:</code> 218, 291 | <code>diag:</code> 1305, 1329 |
| <code>\physicx_xquantity:nn</code> | <code>__physicx_do_nothing:</code> 115 |
| 478, 478, 519, 541, 544, 557, 558 | <code>\l__physicx_end_range_int</code> ... 63, |
| physicx internal commands: | 105, 106, 150, 151, 154, 158, 164, 168 |
| <code>__physicx_adi:nnn</code> 875, 1041, | <code>__physicx_expand:w</code> 854, 885, |
| 1042, 1043, 1044, 1045, 1046, 1858 | 887, 889, 892, 1320, 1431, 1465, 1622 |
| <code>\l__physicx_begin_range_int</code> | <code>__physicx_fixdif:Nn</code> 344, 354 |
| 62, 102, | <code>__physicx_fixdif:Nnn</code> . 344, 348, 356 |
| 103, 148, 149, 153, 154, 158, 164, 168 | <code>\g__physicx_fixdif_bool</code> 261, 311, 330 |
| <code>\l__physicx_cmd_arg_int</code> | <code>__physicx_fixdif_list:N</code> ... 335, 338 |
| 568, 574, 593, 594, 601 | <code>__physicx_generalmatrix:</code> 1739, 1750 |
| <code>\l__physicx_cmd_arg_spec_tl</code> | <code>__physicx_generalmatrix_-</code> |
| 567, 573, 586, 595 | <code>generate:nnn</code> 1754, 1758, 1763 |
| <code>\l__physicx_cmd_auto_body_bool</code> .. | <code>__physicx_if_can_num:n</code> .. 1494, 1494 |
| 566, 576, 621, 664, 668 | <code>__physicx_if_keyval:nTF</code> .. 173, 1529 |
| <code>\l__physicx_cmd_auto_body_tl</code> ... | <code>\l__physicx_invalid_range_bool</code> .. |
| 565, 572, 588, 622, 623 | 66, 85, 95, 140, 155, 165 |
| <code>\l__physicx_cmd_noauto_body_bool</code> | <code>\l__physicx_item_ignore_clist</code> ... |
| 564, 575, 609, 662 | 1419, 1443, 1452 |
| <code>\l__physicx_cmd_noauto_body_tl</code> .. | <code>__physicx_loadpackage_options:nnn</code> |
| 563, 571, 587, 610, 611 | 240, 251, 253, 255 |
| <code>\l__physicx_col_list_seq</code> | <code>\g__physicx_mathtools_bool</code> |
| 852, 906, 1606, 1610 | 9, 26, 281, 282 |
| <code>__physicx_commamatrix_enhanced:</code> | <code>__physicx_matrix_add_data:nn</code> ... |
| 1626, 1655 | 1001, 1007, 1118, 1120 |
| <code>__physicx_commamatrix_enhanced_-</code> | <code>\l__physicx_matrix_after_begin_-</code> |
| <code>aux:nNn</code> 1669, 1677, 1682 | <code>tl</code> 870, 1017, 1019, 1650 |
| <code>__physicx_commamatrix_enhanced_-</code> | <code>\l__physicx_matrix_after_end_tl</code> . |
| <code>aux_e:nnn</code> 1670, 1699 | 871, 1020, 1022, 1652 |
| <code>__physicx_commamatrix_enhanced_-</code> | <code>__physicx_matrix_appto_body_-</code> |
| <code>aux_ne:nnn</code> 1678, 1711 | <code>e:nnn</code> |
| <code>\g__physicx_compat_bool</code> | .. 1164, 1542, 1755, 1811, 1811, 1823 |
| 11, 16, 247, 268, 270 | |

```

\__physicx_matrix_appto_body_-
  ne:nnn .....
  .. 1168, 1546, 1759, 1811, 1824, 1833
\l_physicx_matrix_args_tl .....
  ..... 869, 1015, 1016, 1650
\l_physicx_matrix_array_tl ....
  ..... 843, 882, 1621,
  1622, 1628, 1630, 1633, 1660, 1733
\__physicx_matrix_autocalc:nn ...
  ..... 863, 898, 900, 1416, 1467, 1473
\__physicx_matrix_begin:w .....
  ..... 861, 1012, 1650
\l_physicx_matrix_beginning_tl .
  ..... 876, 1047, 1049, 1857
\l_physicx_matrix_body_tl .....
  ..... 844, 1171, 1206,
  1211, 1212, 1218, 1219, 1224, 1229,
  1236, 1238, 1240, 1242, 1243, 1245,
  1249, 1250, 1255, 1256, 1261, 1267,
  1268, 1270, 1272, 1273, 1568, 1575,
  1584, 1591, 1632, 1644, 1651, 1657,
  1689, 1780, 1789, 1801, 1813, 1826
\__physicx_matrix_calc:nn .....
  ..... 793, 898, 1295
\__physicx_matrix_col_iterate:n .
  860, 993, 1609, 1707, 1717, 1819, 1830
\l_physicx_matrix_cols_int .....
  ..... 797, 798,
  846, 895, 1151, 1190, 1193, 1209,
  1239, 1247, 1269, 1276, 1277, 1331,
  1334, 1338, 1386, 1398, 1409, 1434,
  1487, 1647, 1648, 1839, 1846, 1861,
  1863, 1877, 1883, 1888, 1898, 1904
\l_physicx_matrix_diag_bool ...
  ..... 850, 1303, 1309
\__physicx_matrix_diag_calc:nn ..
  ..... 1294, 1299, 1344,
  1361, 1370, 1379, 1391, 1402, 1415
\l_physicx_matrix_diag_clist ...
  ..... 848, 996, 998, 1526, 1737
\__physicx_matrix_diag_parse_-
  aux:n ..... 1283, 1288
\__physicx_matrix_diag_parse_-
  aux:nn ..... 1284, 1316
\__physicx_matrix_diag_parse_-
  aux_anti:n ..... 1324, 1376
\__physicx_matrix_diag_parse_-
  aux_regu:n ..... 1327, 1350
\l_physicx_matrix_dotcol_bool ..
  ..... 842, 1192, 1196,
  1200, 1216, 1241, 1253, 1271, 1275
\l_physicx_matrix_dotrow_bool ..
  ..... 841, 1177, 1181, 1185, 1233

\__physicx_matrix_element_-
  aux:nnn .... 914, 920, 923, 929,
  932, 938, 941, 947, 950, 956, 959, 964
\__physicx_matrix_end: ... 1013, 1652
\__physicx_matrix_end:w ..... 862
\l_physicx_matrix_ending_tl ...
  ..... 877, 1050, 1052, 1891
\l__physicx_matrix_enhanced_bool
  ..... 873, 1033, 1036, 1537, 1625
\__physicx_matrix_enhanced_init:
  ..... 1598, 1663, 1768
\l__physicx_matrix_expand_-
  element_bool ..... 864,
  972, 1161, 1539, 1602, 1608, 1664, 1752
\__physicx_matrix_generate_-
  body:NNNN .. 1559, 1762, 1765, 1766
\l__physicx_matrix_infinite_bool
  ..... 840, 907, 910, 1148,
  1149, 1223, 1260, 1264, 1728, 1880
\l_physicx_matrix_item_clist ...
  ..... 849, 1002, 1004, 1534, 1738
\__physicx_matrix_item_parse_-
  aux:n ..... 1421, 1426
\__physicx_matrix_item_parse_-
  aux:nn ..... 1422, 1427
\__physicx_matrix_last_aux_c: ...
  ..... 1862, 1884, 1893
\__physicx_matrix_last_aux_r: ...
  ..... 1868, 1885, 1902
\l__physicx_matrix_last_col_tl ..
  ..... 859, 995, 1859, 1873
\l__physicx_matrix_last_row_tl ..
  ..... 858, 994, 1865, 1872
\l__physicx_matrix_main_tl .....
  ..... 847, 902, 1490, 1731
\__physicx_matrix_row_iterate:n .
  857, 992, 1603, 1706, 1716, 1818, 1829
\l__physicx_matrix_rows_int .....
  ..... 795, 796, 845, 894, 1150,
  1175, 1178, 1203, 1225, 1337, 1432,
  1485, 1661, 1692, 1840, 1845, 1867,
  1869, 1876, 1882, 1887, 1895, 1907
\__physicx_matrix_save_or_print:
  ..... 1156, 1554, 1635, 1638, 1740
\l__physicx_matrix_save_tl .....
  ..... 1025, 1027, 1640, 1643
\__physicx_matrix_sep: .....
  ..... 1229, 1236, 1243, 1267,
  1579, 1629, 1695, 1790, 1850, 1850
\l__physicx_matrix_sep_dim .....
  ..... 874, 1023, 1852, 1853
\__physicx_matrix_set_data:nn ...
  ..... 1000, 1006, 1115

```

| | |
|---|---|
| __physicx_matrix_set_r_c- | \l__physicx_quantity_left_tl ... |
| ckall:nnn 832, 990 | 368, 379, 484 |
| __physicx_matrix_set_r_c- | \l__physicx_quantity_post_tl ... |
| ckep:nnn 817, 981 | 369, 378, 517 |
| __physicx_matrix_set_r_c- | \l__physicx_quantity_pre_tl ... |
| ckig:nnn 812, 984 | 370, 377, 492 |
| __physicx_matrix_set_r_c- | \l__physicx_quantity_right_size- |
| ckigep:nnn 823, 833, 987 | tl 371, 495, 500, 508, 513 |
| __physicx_matrix_set_r_c- | \l__physicx_quantity_right_tl ... |
| nock:nnn 810, 835, 978 | 372, 380, 487 |
| \l__physicx_matrix_tmparr_c_sep . | __physicx_quantity_size:nn ... |
| 1472, 1475, 1476 | 381, 382, 396 |
| \l__physicx_matrix_tmparr_r_sep . | __physicx_quantity_unknown:n ... |
| 1466, 1468, 1470 | 394, 405 |
| __physicx_matrix_transpose:N ... | __physicx_qxmatrix_appto- |
| 1551, 1761, 1834, 1834 | body:nnn 1163, 1167, |
| \l__physicx_matrix_transpose- | 1207, 1213, 1220, 1246, 1251, 1257 |
| bool 872, 1028, 1836 | \g__physicx_reqty_bool 13, 257, 674, 749 |
| \l__physicx_max_range_int 64, 81, 105, 138, 150, 151, 153 | \l__physicx_row_list_seq 851, 904, 1600, 1604 |
| \l__physicx_min_range_int 65, 80, 102, 139, 148, 149 | \g__physicx_short_bool ... 12, 21, 249 |
| __physicx_nabla: 333, 336 | \l__physicx_tmp_col_seq .. 1684, 1687 |
| __physicx_nauto_case:nnnn . 637, 660 | \l__physicx_tmp_coled_seq 1686, 1691, 1701, 1713 |
| __physicx_new_matrix_cmd:NNN ... 1911, 1911, 1942, 1943 | \l__physicx_tmp_colnum_seq 1435, 1441, 1450 |
| __physicx_new_xquantity_aux:w .. 525, 531, 535, 539, 542 | \l__physicx_tmp_rownum_seq 1433, 1439, 1448 |
| \g__physicx_original_bool .. 262, 317 | \l__physicx_tmp_seq 1659, 1662, 1666, 1674 |
| __physicx_parse_range_action:nnn 97, 114, 124 | \l__physicx_tmp_tl 596, 610, 622, 1773, 1779, |
| __physicx_parse_range_aux:n 84, 93 | 1782, 1788, 1794, 1800, 1803, 1809 |
| __physicx_parse_range_aux:w ... 115, 116, 125, 126 | \l__physicx_tmpa_bool 51, 193, 200, 205 |
| __physicx_parse_range_range: ... 70, 75, 107, 172 | \l__physicx_tmpa_clist 50 |
| __physicx_parse_range_range- | \l__physicx_tmpa_int 52, 1658, 1668, 1670, 1676, 1678 |
| check: 70, 146, 172 | \l__physicx_tmpa_seq 86, 96, 141, 144, 159, 169 |
| __physicx_parse_range_range- | \l__physicx_tmpa_tl 101, 103, 118, 119, 128, 407, 408 |
| nocheck: 75, 162 | \l__physicx_tmparr_tl 1463, 1464, 1465, 1466 |
| __physicx_parse_range_single:n . 69, 74, 99, 145 | \l__physicx_tmpb_int 53 |
| __physicx_parse_range_single- | \l__physicx_tmpb_tl ... 104, 106, 129 |
| check:n 69, 135, 145 | \l__physicx_tmpdiag_seq 1321, 1332, |
| __physicx_parse_range_single- | 1342, 1345, 1346, 1356, 1362, 1363, |
| nocheck:n 74, 143 | 1365, 1371, 1372, 1380, 1381, 1382, |
| \g__physicx_physics_bool 10 | 1392, 1393, 1394, 1403, 1404, 1405 |
| \l__physicx_quantity_args_tl ... 365, 387, 388, 485 | \l__physicx_tmpdiag_tl 1318, 1319, 1320, 1321 |
| \l__physicx_quantity_code_tl ... 366, 389, 482, 486 | \l__physicx_tmpitem_tl 1429, 1430, 1431, 1437, 1455 |
| \l__physicx_quantity_left_size- | |
| tl 367, 494, 499, 504, 506, 511 | |

| | | | |
|---|--|---|--|
| __physicx_vnabla: | 294, 305, 336 | scan commands: | |
| __physicx_xquantity_aux:nnnn | 483, 490, 520 | \scan_stop: | 1226, 1693 |
| \physicxempty | 806, 820, 828, 865, 866, 974 | seq commands: | |
| \physicxexcept | 868, 945, 954, 963, 969, 971 | \c_empty_seq | 79, 1686 |
| \PHYSICXIGNORE | 7, 8, 814, 825, 1437 | \seq_clear:N | 96 |
| \physicxset | 359 | \seq_concat:NNN | 86 |
| \physicxtmp | 562, 1153, 1522, 1620, 1726 | \seq_count:N | 1345, 1346, 1362, 1363, 1371, 1372, 1380, 1381, 1392, 1393, 1403, 1404, 1468, 1475, 1662 |
| \poissonbracket | 749 | \seq_if_empty:NTF | 1600, 1606 |
| \pqty | 749 | \seq_item:Nn | 1604, 1610 |
| prg commands: | | \seq_map_indexed_inline:Nn | 1332, 1342, 1356, 1365, 1382, 1394, 1405, 1470, 1476, 1687 |
| \prg_generate_conditional_variant:Nnn | 209 | \seq_map_inline:Nn | 1439, 1441, 1448, 1450 |
| \prg_new_conditional:Npnn | 14, 19, 24, 29, 175, 191, 1494 | \seq_map_tokens:Nn | 1666, 1674 |
| \prg_replicate:nn | 1239, 1269 | \seq_new:N | 851, 852 |
| \prg_return_false: | 17, 22, 27, 35, 47, 178, 206, 1515 | \seq_put_right:Nn | 141, 144, 159, 169, 1701, 1713 |
| \prg_return_true: | 17, 22, 27, 34, 47, 178, 206, 1497, 1514 | \seq_set_eq:NN | 79, 1686 |
| \prg_set_conditional:Npnn | 44 | \seq_set_split:Nnn | 5, 904, 906, 1321, 1466, 1472, 1659, 1684 |
| \ProcessKeysPackageOptions | 265 | \seq_use:Nn | 1691 |
| Q | | \setmathfont | 302 |
| \qty | 749 | \setmatrixdata | 1113 |
| \quantity | 674, 752 | \setmatrixtype | 1054, 1076, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1095, 1096, 1097, 1098, 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111 |
| quark commands: | | \setquantitytype | 392, 437, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470 |
| \q_nil | 115 | \smallmatrixquantity | 674 |
| \q_physicx_special | 115, 116, 125, 126 | \smqty | 749 |
| \quark_if_nil:nTF | 120 | str commands: | |
| \quark_if_recursion_tail_stop:n | 630, 631 | \c_backslash_str | 408 |
| \q_recursion_stop | 583 | \str_case_e:nnTF | 1290 |
| \q_recursion_tail | 583 | \str_const:Nn | 837, 838, 839 |
| \qxmatrix | 1135, 1747 | \str_if_eq:nnTF | 615, 617 |
| \qxqty | 561 | \symbf | 224, 294, 336 |
| R | | \symbfit | 220 |
| \rangle | 444, 682, 709, 724 | T | |
| \rbrace | 312, 313 | T _E X and L ^A T _E X 2 _ε commands: | |
| \Re | 275, 312, 322 | \@declareparencmd | 728 |
| \real | 275, 312, 322 | \@declarequantitycmd | 562 |
| regex commands: | | \@dif | 311 |
| \regex_match:nnTF | 177, 954, 963 | \@ifpackageloaded | 267, 280, 290, 307, 320 |
| \renewdif | 335, 336 | \@ifstar | 547 |
| \RequirePackage | 243 | S | |
| \rgroup | 706, 721 | \sb | |
| \right | 227, 475, 500, 699, 706, 709, 715, 716, 721, 723, 724, 725 | 856, 1206, 1211, 1218, 1245, 1249, 1255 | |
| \rVert | 443, 766 | | |
| \rvert | 442 | | |

| | | |
|------------------------------|--|-----------------------------------|
| \@pkgextension | | 46 |
| \bBigg@ | | 238 |
| \c@MaxMatrixCols | | 1032, 1647, 1648 |
| \physicsx@align | | |
| | 837, 1038, 1472, 1631, 1685 | |
| \physicsx@cr | 838, 1037, 1466, 1629, 1659 | |
| \physicsx@matrixelement | | |
| | 856, 911, 915, 916, 924, | |
| | 925, 933, 934, 942, 943, 951, 952, | |
| | 960, 961, 968, 1705, 1715, 1817, 1828 | |
| \physicsx@sep | 839, 904, 906, 1039, 1321 | |
| tex commands: | | |
| \tex_advance:D | | 1276, 1277 |
| \tex_mathclose:D | | 233 |
| \tex_mathopen:D | | 231 |
| text commands: | | |
| \text_expand:n | | 887, 1703, 1815 |
| tl commands: | | |
| \c_empty_tl | | 384, 654 |
| \tl_clear:N | | 571, 572, 573, 1171, 1657 |
| \tl_const:Nn | | 212, 213 |
| \tl_gset_eq:NN | | 1642 |
| \tl_head:N | | 407 |
| \tl_if_blank:nTF | | 130, 927 |
| \tl_if_empty:NTF | | |
| | 101, 104, 1731, 1859, 1865 | |
| \tl_if_empty:nTF | | |
| .. | 112, 482, 819, 827, 918, 1527, 1617 | |
| \tl_if_empty_p:N | | 494, 495, 1872, 1873 |
| \tl_if_eq:nnTF | | 814, 825, 1436 |
| \tl_if_exist:NTF | | 46 |
| \tl_if_head_eq_charcode:nNTF | | 1322 |
| \tl_if_head_eq_meaning:nNTF | | |
| | 350, 357, 503 | |
| \tl_if_head_eq_meaning_p:nN | | |
| | 1501, 1502, 1506, 1511 | |
| \tl_if_in:nnTF | | 174, 936, 945 |
| \tl_if_noalue_p:n | | 599 |
| \tl_map_function:nN | | 335 |
| \tl_new:N | | 365, 366, |
| | 367, 368, 369, 370, 371, 372, 562, | |
| | 563, 565, 567, 843, 844, 847, 858, | |
| | 859, 866, 868, 869, 870, 871, 876, 877 | |
| \tl_put_right:Nn | | |
| | 595, 610, 611, 622, 623, 971, 1019, | |
| | 1022, 1049, 1052, 1206, 1211, 1212, | |
| | 1218, 1219, 1224, 1229, 1236, 1238, | |
| | 1240, 1242, 1243, 1245, 1249, 1250, | |
| | 1255, 1256, 1261, 1267, 1268, 1270, | |
| | 1272, 1273, 1568, 1575, 1584, 1591, | |
| | 1689, 1780, 1789, 1801, 1813, 1826 | |
| \tl_replace_all:Nnn | | 6, 1628, 1630 |
| \tl_set:Nn | | 118, 119, |
| | 128, 129, 387, 400, 407, 482, 596, | |
| | 811, 815, 820, 821, 828, 829, 1015, | |
| | 1027, 1318, 1319, 1429, 1430, 1463, | |
| | 1464, 1621, 1773, 1782, 1794, 1803 | |
| \tl_set_eq:NN | | 286, 287, 403, 1632 |
| \tl_tail:N | | 411, 414 |
| \tl_tail:n | | 351, 357, 1325 |
| \tl_trim_spaces:n | .. | 118, 119, 128, 129 |
| token commands: | | |
| \token_if_cs:NTF | | 1640 |
| \token_if_eq_meaning:NNTF | | 408 |
| \token_if_eq_meaning_p:NN | .. | 499, 500 |
| \TrimSpaces | | 437, 1080 |
| | | |
| | | U |
| use commands: | | |
| \use:N | | 401 |
| \use:n | | 41, 409, 417, 892 |
| \use:nn | | 1765, 1847 |
| \use:nnnn | | 4 |
| \use_i:nn | | 40, 638, 1433, 1550, 1746 |
| \use_i:nnn | | 1746 |
| \use_ii:nnn | | 1548, 1745 |
| \use_ii:nn | | 409, 638, 1435 |
| \use_iii:nn | | 1765, 1841 |
| \use_none:n | | 42, 415 |
| \use_none:nn | | 863, 900, 1299 |
| \use_none:nnn | | 37 |
| | | |
| | | V |
| \vdots | | 936, 1238, |
| | 1240, 1242, 1268, 1270, 1272, 1908 | |
| \vec | | 299 |
| \vectimes | | 304 |
| \vectorarrow | | 298 |
| \vectorbold | | 296 |
| \vectorunit | | 300 |
| \Vert | | 683, 725 |
| \vert | | 681, 688, 689, 690, 723, 755, 759 |
| \vnabla | | 305 |
| \vqty | | 749 |
| \vysmbkcircle | | 303 |
| | | |
| | | X |
| \xquantity | | 519 |
| | | |
| | | Z |
| \Z | | 177 |