

The `physicx` package

Wenjian Chern (Longaster*)

February 2, 2023, version v0.3.6

Abstract

`physicx`

1 Implementation

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

*Email: longaster@163.com

```

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 \d }
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 \AtBeginDocument
333 {
334 \@ifpackageloaded { unicode-math }
335 { \exp_args:NNNx \renewdef * \_physics_vnabla: { \exp_not:o \_physics_vnabla: }
336 {
337 \cs_if_exist:NT \vnabla
338 {
339 \cs_set_eq:NN \_physics_vnabla: \vnabla
340 \renewdef * \vnabla { \_physics_vnabla: }
341 \cs_set_protected:Npx \vnabla { \exp_not:o \vnabla }
342 }
343 }
344 }
345 }

```

`\physicsset` physics setup command.

```

346 \NewDocumentCommand \physicsset { s m }
347 {

```

```

348     \IfBooleanTF {#1}
349     { \keys_set:nn { physicx/#2 } }
350     { \keys_set:nn { physicx } {#2} }
351 }

```

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

1.2 Quantity things

1.2.1 New quantity interfaces

```

352 \tl_new:N \l__physicx_quantity_args_tl
353 \tl_new:N \l__physicx_quantity_code_tl
354 \tl_new:N \l__physicx_quantity_left_size_tl
355 \tl_new:N \l__physicx_quantity_left_tl
356 \tl_new:N \l__physicx_quantity_post_tl
357 \tl_new:N \l__physicx_quantity_pre_tl
358 \tl_new:N \l__physicx_quantity_right_size_tl
359 \tl_new:N \l__physicx_quantity_right_tl
360 \keys_define:nn { physicx }
361 { quantity .code:n = \keys_set:nn { physicx/quantity } {#1} }
362 \keys_define:nn { physicx/quantity }
363 {
364   pre .tl_set:N = \l__physicx_quantity_pre_tl ,
365   post .tl_set:N = \l__physicx_quantity_post_tl ,
366   left .tl_set:N = \l__physicx_quantity_left_tl ,
367   right .tl_set:N = \l__physicx_quantity_right_tl ,
368   left-size .code:n = { \__physicx_quantity_size:nn { left } {#1} } ,
369   right-size .code:n = { \__physicx_quantity_size:nn { right } {#1} } ,
370   size .meta:n = { left-size = {#1} , right-size = {#1} } ,
371   auto .meta:n = { left-size = \left , right-size = \right } ,
372   noauto .meta:n = { left-size = \c_empty_tl , right-size = \c_empty_tl } ,
373   noauto .value_required:n = false ,
374   args .code:n =
375     \tl_set:Nn \l__physicx_quantity_args_tl { [#1] } ,
376   args* .tl_set:N = \l__physicx_quantity_args_tl ,
377   code .tl_set:N = \l__physicx_quantity_code_tl ,
378   type .multichoice: ,
379
380   settype .code:n = \setquantitytype #1 ,
381
382   unknown .code:n = \__physicx_quantity_unknown:n {#1} ,
383 }
384 \cs_new:Npn \__physicx_quantity_size:nn #1#2
385 {
386   \physicx_if_num:nTF {#2}
387   {
388     \tl_set:cx { l__physicx_quantity_ #1 _size_tl }
389     { \use:c { physicx_ #1:nN } { \fp_eval:n {#2} } }
390   }
391   { \tl_set_eq:cN { l__physicx_quantity_ #1 _size_tl } #2 }
392 }
393 \cs_new:Npn \__physicx_quantity_unknown:n #1
394 {
395   \int_compare:nNnTF { \c_zero_int } =

```



```

396 { \char_value_catcode:n { \exp_last_unbraced:Ne ' { \tl_head:N \l_keys_key_str } } }
397 { \use:n } { \use_ii:nn }
398 {
399   \cs_if_exist:cTF { \tl_tail:N \l_keys_key_str }
400   {
401     \keys_set:nx { physicx/quantity }
402     { size = \exp_not:c { \tl_tail:N \l_keys_key_str } }
403     \use_none:n
404   }
405   { \use:n }
406 }
407 {
408   \exp_args:No \physicx_if_num:nTF \l_keys_key_str
409   {
410     \keys_set:nx { physicx/quantity } { size = \l_keys_key_str }
411   }
412   {
413     \physicx_search_also:nnF
414     {
415       physicx/quantity/type ,
416     }
417     {#1}
418     {
419       \msg_error:nxxx { physicx } { unknown-key }
420       \l_keys_path_str { physicx/quantity }
421     }
422   }
423 }
424 }
425 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
426 { \physicx_new_type:nnn { quantity } {#1} }
427 \setquantitytype { b } { left={[] , right=[]} , }
428 \setquantitytype { B } { left={\{ , right={\}} , }
429 \setquantitytype { p } { left={() , right={)} , }
430 \setquantitytype { v } { left=\lvert , right=\rvert , }
431 \setquantitytype { V } { left=\lVert , right=\rVert , }
432 \setquantitytype { a } { left=\langle , right=\rangle , }
433 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
434 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
435 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
436 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
437 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
438 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
439 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
440 \physicx_mathtools:T
441 {
442   \setquantitytype { m* } { left=\begin{matrix*} , right=\end{matrix*} , noauto }
443   \setquantitytype { bm* } { left=\begin{bmatrix*} , right=\end{bmatrix*} , noauto }
444   \setquantitytype { Bm* } { left=\begin{Bmatrix*} , right=\end{Bmatrix*} , noauto }
445   \setquantitytype { pm* } { left=\begin{pmatrix*} , right=\end{pmatrix*} , noauto }
446   \setquantitytype { vm* } { left=\begin{vmatrix*} , right=\end{vmatrix*} , noauto }
447   \setquantitytype { Vm* } { left=\begin{Vmatrix*} , right=\end{Vmatrix*} , noauto }
448   \setquantitytype { sm* } { left=\begin{smallmatrix*} , right=\end{smallmatrix*} , noauto }
449   \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }

```

```

450 \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto
451 \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto
452 \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto
453 \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto
454 \setquantitytype { sbm* } { left=\begin{bsmallmatrix*} , right=\end{bsmallmatrix*} , noa
455 \setquantitytype { sBm* } { left=\begin{Bsmallmatrix*} , right=\end{Bsmallmatrix*} , noa
456 \setquantitytype { spm* } { left=\begin{psmallmatrix*} , right=\end{psmallmatrix*} , noa
457 \setquantitytype { svm* } { left=\begin{vsmallmatrix*} , right=\end{vsmallmatrix*} , noa
458 \setquantitytype { sVm* } { left=\begin{Vsmallmatrix*} , right=\end{Vsmallmatrix*} , noa
459 }
460 \keys_set:nn { physicx/quantity }
461 {
462 left-size = \left ,
463 right-size = \right ,
464 type = p ,
465 }

\physicx_xquantity:nn
\newxquantity
\NewXQuantity
466 \cs_new:Npn \physicx_xquantity:nn #1#2
467 {
468 \group_begin:
469 \keys_set:nn { physicx/quantity } {#1}
470 \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
471 \__physicx_xquantity_aux:oooo
472 { \l__physicx_quantity_left_tl }
473 { \l__physicx_quantity_args_tl }
474 { \l__physicx_quantity_code_tl }
475 { \l__physicx_quantity_right_tl }
476 \group_end:
477 }
478 \cs_new:Npn \__physicx_xquantity_aux:nnnn #1#2#3#4
479 {
480 \l__physicx_quantity_pre_tl
481 \bool_lazy_or:nnTF
482 { \tl_if_empty_p:N \l__physicx_quantity_left_size_tl }
483 { \tl_if_empty_p:N \l__physicx_quantity_right_size_tl }
484 { #1 #2 #3 #4 }
485 {
486 \bool_lazy_or:nnTF
487 { \token_if_eq_meaning_p:NN \l__physicx_quantity_left_size_tl \left }
488 { \token_if_eq_meaning_p:NN \l__physicx_quantity_right_size_tl \right }
489 { \physicx_left: #1 #2 #3 \physicx_right: #4 }
490 {
491 \exp_args:No \tl_if_head_eq_meaning:nNTF
492 \l__physicx_quantity_left_size_tl \physicx_left:nN
493 {
494 \l__physicx_quantity_left_size_tl #1 #2
495 #3
496 \l__physicx_quantity_right_size_tl #4
497 }
498 {
499 \physicx_left:N \l__physicx_quantity_left_size_tl #1 #2
500 #3
501 \physicx_right:N \l__physicx_quantity_right_size_tl #4

```

```

502     }
503   }
504 }
505 \l__physicx_quantity_post_tl
506 }
507 \NewDocumentCommand \xquantity { } { \physicx_xquantity:nn }
508 \cs_generate_variant:Nn \__physicx_xquantity_aux:nnnn { oooo }
509 \NewDocumentCommand \newxquantity { m o o m m }
510 {
511   \IfNoValueTF {#2}
512   {
513     \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
514       { \newcommand ##1 }
515   }
516   {
517     \IfNoValueTF {#3}
518     {
519       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
520         { \newcommand ##1 [#2] }
521     }
522     {
523       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
524         { \newcommand ##1 [#2] [#3] }
525     }
526   }
527   \exp_args:Nc \__physicx_new_xquantity_aux:w
528   { \cs_to_str:N #1~star }
529   { \physicx_xquantity:nn { #4 , noauto } {#5} }
530   \exp_args:Nc \__physicx_new_xquantity_aux:w
531   { \cs_to_str:N #1~unstar }
532   { \physicx_xquantity:nn { #4 } {#5} }
533   \exp_args:NNx \newcommand #1
534   {
535     \exp_not:N \@ifstar
536     \exp_not:c { \cs_to_str:N #1~star }
537     \exp_not:c { \cs_to_str:N #1~unstar }
538   }
539 }
540 \NewDocumentCommand \NewXQuantity { m m m m }
541 {
542   \NewDocumentCommand #1 { s #2 }
543   {
544     \IfBooleanTF {##1}
545     { \physicx_xquantity:nn { #3 , noauto } {#4} }
546     { \physicx_xquantity:nn { #3 } {#4} }
547   }
548 }
549 \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

```

550 \tl_new:N \physicx_tmp
551 \tl_new:N \l__physicx_cmd_noauto_body_tl
552 \bool_new:N \l__physicx_cmd_noauto_body_bool
553 \tl_new:N \l__physicx_cmd_auto_body_tl
554 \bool_new:N \l__physicx_cmd_auto_body_bool
555 \tl_new:N \l__physicx_cmd_arg_spec_tl
556 \int_new:N \l__physicx_cmd_arg_int
557 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
558 {
559   \tl_clear:N \l__physicx_cmd_noauto_body_tl
560   \tl_clear:N \l__physicx_cmd_auto_body_tl
561   \tl_clear:N \l__physicx_cmd_arg_spec_tl
562   \int_set:Nn \l__physicx_cmd_arg_int {#1}
563   \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
564   \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
565 }
566 % noauto, auto, cmd, body
567 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
568 {
569   \__physicx_declare_init:nnn { 3 } {#1} {#2}
570   \__physicx_declare_legacy_quantity_aux:nw #4
571   \q_recursion_tail \q_recursion_tail \q_recursion_stop
572   \__physicx_declare_legacy_quantity_aux:NcVVV
573   #3 { \cs_to_str:N #3 ~ body }
574   \l__physicx_cmd_arg_spec_tl
575   \l__physicx_cmd_noauto_body_tl
576   \l__physicx_cmd_auto_body_tl
577 }
578 % arg spec, pre, body to replace(start from #4), post
579 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
580 {
581   \int_incr:N \l__physicx_cmd_arg_int
582   \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
583     \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
584     \tl_set:Nx \l__physicx_tmp_tl
585     {
586       {
587         \exp_not:N \tl_if_novalue_p:n
588         {
589           \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
590           \or: \or: \or:
591           \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
592           \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
593           \fi:
594         }
595       }
596     }
597   \if_bool:N \l__physicx_cmd_noauto_body_bool
598     \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
599     \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
600     {

```

```

601         {
602             % if is '.', use none
603             \str_if_eq:nnTF {#2} {.} {} {#2}
604             #3
605             \str_if_eq:nnTF {#4} {.} {} {#4}
606         }
607     }
608     \fi:
609     \if_bool:N \l__physicx_cmd_auto_body_bool
610         \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
611         \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
612         { { ##1 #2 #3 ##2 #4 } }
613     \fi:
614     \fi:
615 }
616 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
617 {
618     \quark_if_recursion_tail_stop:n {#1}
619     \quark_if_recursion_tail_stop:n {#2}
620     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
621     \__physicx_declare_legacy_quantity_aux:nw
622 }
623 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
624 {
625     \__physicx_nauto_case:nnnn
626     { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
627     {
628         \cs_set_protected:Npn #1
629         {
630             \peek_charcode_ignore_spaces:NTF \let
631             { #2 } { #2 [ \physicx_left: ] \physicx_right: }
632         }
633         \DeclareDocumentCommand #2 { 0{##2} m s #3 }
634         {
635             \IfBooleanTF { ##3 }
636             { \bool_case_false:n {#4} }
637             { \bool_case_false:n {#5} }
638         }
639     }
640     {
641         \cs_set_protected:Npn #1
642         { #2 \c_empty_tl \c_empty_tl }
643         \DeclareDocumentCommand #2 { m m s #3 }
644         { \bool_case_false:n {#4} }
645     }
646 }
647 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
648 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
649 {
650     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
651     {
652         \bool_if:NTF \l__physicx_cmd_auto_body_bool
653         {#1} {#2}
654     }

```

```

655     {
656       \bool_if:NTF \l__physicsx_cmd_auto_body_bool
657       {#3} {#4}
658     }
659   }
660 \cs_set_protected:Npn \@declarequantitycmd
661 { \physicsx_declare_legacy_quantity:nnNn }

```

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

```

\quantity      Redefine some macros in physics package.
\evaluated
\matrixquantity
\smallmatrixquantity
662 \if_bool:N \g__physicsx_reqty_bool
663 \physicsx_declare_legacy_quantity:nnNn
664 \c_true_bool \c_true_bool \quantity
665 {
666   { !g } { { \{ } { #4 } { \} } }
667   { !o } { { [ } { #5 } { ] } }
668   { !d() } { { ( } { #6 } { ) } }
669   { !d|| } { { \vert } { #7 } { \vert } }
670   { !d<> } { { \langle } { #8 } { \rangle } }
671   { !d== } { { \Vert } { #9 } { \Vert } }
672 }
673 \physicsx_declare_legacy_quantity:nnNn
674 \c_true_bool \c_true_bool \evaluated
675 {
676   { !g } { { . } { #4 \nobreak } { \vert } }
677   { !d[] } { { [ } { #5 \nobreak } { \vert } }
678   { !d( ) } { { ( } { #6 \nobreak } { \vert } }
679 }
680 \physicsx_declare_legacy_quantity:nnNn
681 \c_true_bool \c_false_bool \matrixquantity
682 {
683   { !g }
684   {
685     { \IfBooleanT{#3}{\left\{ } }
686     { \begin{matrix} #4 \end{matrix} }
687     { \IfBooleanT{#3}{\right\} }
688   }
689   { !o } { { \begin{bmatrix} } { #5 } { \end{bmatrix} } }
690   { !d() }
691   {
692     { \IfBooleanTF{#3}{\left\lgroup}{\left( } }
693     { \begin{matrix} #6 \end{matrix} }
694     { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
695   }
696   { !d|| } { { \begin{vmatrix} } { #7 } { \end{vmatrix} } }
697   { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
698   { !d== } { { \begin{Vmatrix} } { #9 } { \end{Vmatrix} } }
699 }
700 \physicsx_declare_legacy_quantity:nnNn
701 \c_true_bool \c_false_bool \smallmatrixquantity
702 {
703   { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }

```

```

704 { !o } { {\left[] { \begin{smallmatrix} #5 \end{smallmatrix} } {\right[] } }
705 { !d() }
706 {
707   { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
708   { \begin{smallmatrix} #6 \end{smallmatrix} } }
709   { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
710 }
711 { !d|| } { {\left\vert} { \begin{smallmatrix} #7 \end{smallmatrix} } {\right\vert} }
712 { !d<> } { {\left\langle} { \begin{smallmatrix} #8 \end{smallmatrix} } {\right\rangle} }
713 { !d== } { {\left\Vert} { \begin{smallmatrix} #9 \end{smallmatrix} } {\right\Vert} }
714 }
715 \fi:

```

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

\physics_declare_legacy_paren:NnnnNn
 \@declareparencmd

```

716 %% cmd, arg spec, replace(start from #6), pre, left, right, post
717 \cs_new:Npn \physics_declare_legacy_paren:NnnnNn #1#2#3#4#5#6#7
718 {
719   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
720   {
721     \bool_case_true:nF
722     {
723       { \bool_if_p:n {##2} } { #4 \physics_left:N \bigl #5 #3 \physics_right:N \bigr
724       { \bool_if_p:n {##3} } { #4 \physics_left:N \Bigl #5 #3 \physics_right:N \Bigr
725       { \bool_if_p:n {##4} } { #4 \physics_left:N \biggl #5 #3 \physics_right:N \biggr
726       { \bool_if_p:n {##5} } { #4 \physics_left:N \Biggl #5 #3 \physics_right:N \Biggr
727     }
728     {
729       \IfBooleanTF {##1}
730       { #4 #5 #3 #6 #7 }
731       { #4 \physics_left: #5 #3 \physics_right: #6 #7 }
732     }
733   }
734 }
735 \cs_set_protected:Npn \@declareparencmd
736 { \physics_declare_legacy_paren:NnnnNn }

```

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

```

\qty Redefine some macros in physics package.
\mqty 737 \if_bool:N \g__physics_reqty_bool
\smqty 738 \physics_option_or:nnT { compat } { short }
\pqty 739 {
\bqty 740   \cs_set:Npn \qty { \quantity }
\vqty 741   \physics_declare_legacy_paren:NnnnNn \pqty { m } {#6} { } { } { }
\Bqty 742   \physics_declare_legacy_paren:NnnnNn \bqty { m } {#6} { } { } { }
\absolutevalue 743   \physics_declare_legacy_paren:NnnnNn \vqty { m } {#6} { } { } \vert \vert { }
\eval 744   \physics_declare_legacy_paren:NnnnNn \Bqty { m } {#6} { } { } \{ \} { }
\abs 745 }
\norm 746 \physics_declare_legacy_paren:NnnnNn \absolutevalue
\order 747 { m } {#6} { } { } \vert \vert { }
\oorder 748 \physics_option_or:nnT { compat } { short }
\commutator 749 {
\poissonbracket
\pb
\anticommutator
\acomm

```

```

750     \cs_set:Npn \eval { \evaluated }
751     \cs_set:Npn \abs { \absolutevalue }
752 }
753 \physicsx_declare_legacy_paren:NnnnNNn \norm
754 { m } {#6} { } \lVert \rVert { }
755 \physicsx_compat:TF
756 {
757     \physicsx_declare_legacy_paren:NnnnNNn \order
758     { m } {#6} { \c_physicsx_Order_tl } ( ) { }
759 }
760 {
761     \physicsx_declare_legacy_paren:NnnnNNn \order
762     { m } {#6} { \c_physicsx_order_tl } ( ) { }
763 }
764 \physicsx_declare_legacy_paren:NnnnNNn \commutator
765 { m m } { #6 , #7 } { } [ ] { }
766 \physicsx_option_or:nnT { compat } { short }
767 { \cs_set:Npn \comm { \commutator } }
768 \physicsx_declare_legacy_paren:NnnnNNn \poissonbracket
769 { m m } { #6 , #7 } { } \{ \} { }
770 \physicsx_option_or:nnT { compat } { short }
771 {
772     \cs_set:Npn \pb { \poissonbracket }
773     \cs_set:Npn \anticommutator { \poissonbracket }
774     \cs_set:Npn \acomm { \poissonbracket }
775 }
776 \fi:
777 \physicsx_declare_legacy_paren:NnnnNNn \OOrder
778 { m } {#6} { \c_physicsx_Order_tl } ( ) { }
779 \physicsx_declare_legacy_paren:NnnnNNn \oorder
780 { m } {#6} { \c_physicsx_order_tl } ( ) { }

```

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

1.3 Matrix things

1.3.1 Matrix auxillary functions

```

781 \cs_new_nopar:Npn \__physicsx_matrix_calc:nn #1#2
782 {
783     \int_set:Nn \l__physicsx_matrix_rows_int
784     { \int_max:nn {#1} \l__physicsx_matrix_rows_int }
785     \int_set:Nn \l__physicsx_matrix_cols_int
786     { \int_max:nn {#2} \l__physicsx_matrix_cols_int }
787 }
788 % use matrix element
789 \cs_new_nopar:Npn \physicsx_matrix_use_r_c:nn #1#2
790 {
791     \if_cs_exist:w l__physicsx_matrix_r@#1_c@#2_tl \cs_end:
792     \exp_not:v { l__physicsx_matrix_r@#1_c@#2_tl }
793     \else:
794     \exp_not:o { \physicsxempty }
795     \fi:
796 }
797 % set matrix element, check or not

```



```

798 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
799   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
800 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
801   {
802     \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
803     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
804   }
805 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
806   {
807     \tl_if_empty:nTF {#3}
808     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
809     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
810   }
811 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
812   {
813     \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
814     {
815       \tl_if_empty:nTF {#3}
816       { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
817       { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
818     }
819   }
820 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
821   \__physicx_matrix_set_r_c_ckigep:nnn
822 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
823   \__physicx_matrix_set_r_c_nock:nnn
824 % align, cr, sep symbol
825 \str_const:Nn \physicx@align { , }
826 \str_const:Nn \physicx@cr { ; }
827 \str_const:Nn \physicx@sep { , }
828 \bool_new:N \l__physicx_matrix_infinite_bool
829 \bool_new:N \l__physicx_matrix_dotrow_bool
830 \bool_new:N \l__physicx_matrix_dotcol_bool
831 \tl_new:N \l__physicx_matrix_array_tl
832 \tl_new:N \l__physicx_matrix_body_tl
833 \int_new:N \l__physicx_matrix_rows_int
834 \int_new:N \l__physicx_matrix_cols_int
835 \tl_new:N \l__physicx_matrix_main_tl
836 \clist_new:N \l__physicx_matrix_diag_clist
837 \clist_new:N \l__physicx_matrix_item_clist
838 \bool_new:N \l__physicx_matrix_diag_bool
839 \seq_new:N \l__physicx_row_list_seq
840 \seq_new:N \l__physicx_col_list_seq
841 % expand input
842 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
843 %% main, row iterate, col iterate
844 \cs_new_nopar:Npn \physicx@matricelement #1#2#3 { #1 \sb { #2 #3 } }
845 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
846 \tl_new:N \l__physicx_matrix_last_row_tl
847 \tl_new:N \l__physicx_matrix_last_col_tl
848 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
849 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
850 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
851 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn

```

```

852 \bool_new:N \l__physicx_matrix_expand_element_bool
853 % when element is empty use \physicxempty
854 \tl_new:N \physicxempty
855 % save 'element-except' key's value
856 \tl_new:N \physicxexcept
857 \tl_new:N \l__physicx_matrix_args_tl
858 \tl_new:N \l__physicx_matrix_after_begin_tl
859 \tl_new:N \l__physicx_matrix_after_end_tl
860 \bool_new:N \l__physicx_matrix_transpose_bool
861 \bool_new:N \l__physicx_matrix_enhanced_bool
862 \dim_new:N \l__physicx_matrix_sep_dim
863 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
864 \tl_new:N \l__physicx_matrix_beginning_tl
865 \tl_new:N \l__physicx_matrix_ending_tl

```

1.3.2 Matrix keys

```

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

```

```

904 \cs_set:Npn \physicsx@matricelement ##1##2##3
905 {
906   \tl_if_empty:nTF {##1}
907     {##1}
908     { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
909   } ,
910 element-code* / except-blank .code:n =
911   \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
912     \physicsx@matricelement
913   \cs_set:Npn \physicsx@matricelement ##1##2##3
914   {
915     \tl_if_blank:nTF {##1}
916       {##1}
917       { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
918   } ,
919 element-code* / except-dots .code:n =
920   \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
921     \physicsx@matricelement
922   \cs_set:Npn \physicsx@matricelement ##1##2##3
923   {
924     \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
925       {##1}
926       { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
927   } ,
928 element-code* / except-tl .code:n =
929   \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
930     \physicsx@matricelement
931   \cs_set:Npn \physicsx@matricelement ##1##2##3
932   {
933     \tl_if_in:onTF { \physicsxexcept } {##1}
934       {##1}
935       { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
936   } ,
937 element-code* / except-regex .code:n =
938   \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
939     \physicsx@matricelement
940   \cs_set:Npn \physicsx@matricelement ##1##2##3
941   {
942     \exp_args:No \regex_match:nnTF { \physicsxexcept } {##1}
943       {##1}
944       { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
945   } ,
946 element-code* / only-regex .code:n =
947   \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
948     \physicsx@matricelement
949   \cs_set:Npn \physicsx@matricelement ##1##2##3
950   {
951     \exp_args:No \regex_match:nnTF { \physicsxexcept } {##1}
952       { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
953       {##1}
954   } ,
955 element-code* / unknown .code:n =
956   \cs_set:Npx \physicsx@matricelement { \exp_not:c {#1} } ,
957 element-except .tl_set:N = \physicsxexcept ,

```

```

958 element-except+ .code:n =
959   \tl_put_right:Nn \physicsexcept {#1} ,
960   expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
961   expand-element .default:n = true ,
962   empty .tl_set:N = \physicxempty ,
963   check .choice: ,
964   check / none .code:n =
965     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
966     \__physicx_matrix_set_r_c_nock:nnn ,
967   check / empty .code:n =
968     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
969     \__physicx_matrix_set_r_c_cke:nnn ,
970   check / ignore .code:n =
971     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
972     \__physicx_matrix_set_r_c_ckig:nnn ,
973   check / igep .code:n =
974     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
975     \__physicx_matrix_set_r_c_ckigep:nnn ,
976   check / all .code:n =
977     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
978     \__physicx_matrix_set_r_c_ckall:nnn ,
979   check .default:n = all ,
980   row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
981   col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
982   last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
983   last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
984   diag .clist_set:N = \l__physicx_matrix_diag_clist ,
985   diag+ .code:n =
986     \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
987   diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
988   diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
989   diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
990   item .clist_set:N = \l__physicx_matrix_item_clist ,
991   item+ .code:n =
992     \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
993   item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
994   item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
995   item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
996   check-range .choice: ,
997   check-range / true .code:n = \physicx_parse_range_check: ,
998   check-range / false .code:n = \physicx_parse_range_noccheck: ,
999   check-range .default:n = true ,
1000   begin .tl_set:N = \__physicx_matrix_begin:w ,
1001   end .tl_set:N = \__physicx_matrix_end: ,
1002   args .code:n =
1003     \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
1004   args* .tl_set:N = \l__physicx_matrix_args_tl ,
1005   after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
1006   after-begin+ .code:n =
1007     { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
1008   after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
1009   after-end+ .code:n =
1010     { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
1011   sepdim .dim_set:N = \l__physicx_matrix_sep_dim ,

```

```

1012     type .multichoice: ,
1013     saveto .tl_set:N = \l__physicx_matrix_save_tl ,
1014     saveto* .code:n =
1015         \tl_set:N \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
1016     transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
1017     transpose .default:n = true ,
1018     ' .meta:n = { transpose = true } ,
1019     T .meta:n = { transpose = true } ,
1020     MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
1021     enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
1022     enhanced .default:n = true ,
1023     !enhanced .code:n =
1024         \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
1025     cr .tl_set:N = \physicx@cr ,
1026     align .tl_set:N = \physicx@align ,
1027     sep .tl_set:N = \physicx@sep ,
1028     adi-order .choice: ,
1029     adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
1030     adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
1031     adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
1032     adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
1033     adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
1034     adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
1035     beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
1036     beginning+ .code:n =
1037         \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
1038     ending .tl_set:N = \l__physicx_matrix_ending_tl ,
1039     ending+ .code:n =
1040         \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
1041
1042     settype .code:n = \setmatrixtype #1 ,
1043
1044     unknown .code:n =
1045         \physicx_search_also:nnF
1046         {
1047             physicx/matrix/type ,
1048             physicx/matrix/expand ,
1049             physicx/matrix/element-code* ,
1050         }
1051         {#1}
1052         {
1053             \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
1054             {
1055                 \keys_set:nx { physicx/matrix }
1056                 { MaxMatrixCols = \l_keys_key_str }
1057             }
1058             {
1059                 \msg_error:nnxx { physicx } { unknown-key }
1060                 \l_keys_path_str { physicx/matrix }
1061             }
1062         } ,
1063 }

```

```

\physicx_matrix_new_type:nnn
\physicx_matrix_new_type:nn
\setmatrixtype

```

```

1064 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
1065 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
1066 \cs_new:Npn \physicx_matrix_new_type:nn
1067 { \physicx_new_type:nnn { matrix } }
1068 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
1069 {
1070   \IfBooleanTF {#1}
1071   { \physicx_matrix_new_type:nn {#2} }
1072   { \physicx_matrix_new_type:nnn {#2} }
1073 }

```

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

A few types.

```

1074 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
1075 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
1076 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
1077 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
1078 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
1079 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
1080 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
1081 \physicx_mathtools:T
1082 {
1083   \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
1084   \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
1085   \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
1086   \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
1087   \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
1088   \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
1089   \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
1090   \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
1091   \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
1092   \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
1093   \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
1094   \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
1095   \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
1096   \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
1097   \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
1098   \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
1099   \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
1100 }

```

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

```

1101 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
1102 { \clist_set:cn { physicx@ #1 data@ #2 } }
1103 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
1104 {
1105   \clist_clear:c { l__physicx_matrix_ #1 _clist }
1106   \__physicx_matrix_add_data:nn {#1} {#2}
1107 }
1108 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
1109 {
1110   \clist_map_inline:nn {#2}
1111   {

```

```

1112         \clist_concat:ccc
1113         { l__physicx_matrix_ #1 _clist }
1114         { l__physicx_matrix_ #1 _clist }
1115         { physicx@ #1 data@ #2 }
1116     }
1117 }

```

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

Initial settings.

```

1118 \keys_set:nn { physicx/matrix }
1119 {
1120     type = m ,
1121     saveto = ? ,
1122 }

```

\qxmatrix

```

1123 %% basically, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
1124 xmatmmn-in-the-physics-package, but changed some
1125 % #1 = boolean, saveto matrix
1126 % #2 = star, infinite
1127 % #3 = options
1128 % #4 = letter for the entries
1129 % #5 = number of rows
1130 % #6 = number of explicit rows, default = 3
1131 % #7 = number of columns
1132 % #8 = number of explicit columns, default = 3
1133 \DeclareDocumentCommand \qxmatrix { t= s 0{type=p} m m 0{3} m 0{3} }
1134 {
1135     \group_begin:
1136     \IfBooleanTF { #2 }
1137     { \bool_set_true:N \l__physicx_matrix_infinite_bool }
1138     { \bool_set_false:N \l__physicx_matrix_infinite_bool }
1139     \int_set:Nn \l__physicx_matrix_rows_int {#6}
1140     \int_set:Nn \l__physicx_matrix_cols_int {#8}
1141     \IfBooleanTF {#1}
1142     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1143     { \keys_set:nn { physicx/matrix } {#3} }
1144     \physicx_qxmatrix:nnn {#4} {#5} {#7}
1145     \__physicx_matrix_save_or_print:
1146     \group_end:
1147 }
1148 \cs_new_protected:Nn \physicx_qxmatrix:nnn
1149 {
1150     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1151     {
1152         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1153         \__physicx_matrix_appto_body_e:nnn
1154     }
1155     {
1156         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1157         \__physicx_matrix_appto_body_ne:nnn
1158     }
1159     % clear the variable containing the body of the matrix
1160     \tl_clear:N \l__physicx_matrix_body_tl

```

```

1160 % set the tentative number of explicit rows
1161 \physicx_if_num:nTF { #2 }
1162 {
1163   {% number of rows is an integer
1164     \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1165     {% if #2 <= rows, we don't want a row of dots
1166       \bool_set_false:N \l__physicx_matrix_dotrow_bool
1167       \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1168     }
1169     {% we want a row of dots
1170       \bool_set_true:N \l__physicx_matrix_dotrow_bool
1171     }
1172   }
1173   {% number of rows is symbolic, we want a row of dots
1174     \bool_set_true:N \l__physicx_matrix_dotrow_bool
1175   }
1176 % set the tentative number of explicit columns
1177 \physicx_if_num:nTF { #3 }
1178 {
1179   {% number of cols is an integer
1180     \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
1181     {% if #3 <= cols, we don't want a column of dots
1182       \bool_set_false:N \l__physicx_matrix_dotcol_bool
1183       \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1184     }
1185     {% we want a column of dots
1186       \bool_set_true:N \l__physicx_matrix_dotcol_bool
1187     }
1188   }
1189   {% number of columns is symbolic, we want a column of dots
1190     \bool_set_true:N \l__physicx_matrix_dotcol_bool
1191   }
1192 % loop through the rows
1193 \int_step_inline:nn { \l__physicx_matrix_rows_int }
1194 {
1195   % add the first entry in the row
1196   %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1197   \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1198   % add the further entries in the explicit columns
1199   \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1200   {
1201     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1202     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1203     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1204   }
1205   % if we have a column of dots, add \cdots and the last entry
1206   \bool_if:NT \l__physicx_matrix_dotcol_bool
1207   {
1208     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1209     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1210     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1211   }
1212   % infinite matrix, add \cdots
1213   \bool_if:NT \l__physicx_matrix_infinite_bool
1214   { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1215   \if_int_compare:w ##1 = \l__physicx_matrix_rows_int

```



```

1214         \scan_stop:
1215     \else:
1216         % finish up the row
1217         \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1218     \fi:
1219 }
1220 % finish up the rows
1221 \bool_if:NT \l__physicx_matrix_dotrow_bool
1222 {
1223     % finish up the row
1224     \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1225     % if we have a row of dots, fill it in
1226     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1227     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1228     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1229     \bool_if:NT \l__physicx_matrix_dotcol_bool
1230     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1231     \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1232     % fill the last row
1233     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1234     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1235     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1236     {
1237         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1238         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1239         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1240     }
1241     \bool_if:NT \l__physicx_matrix_dotcol_bool
1242     {
1243         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1244         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1245         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1246     }
1247     % if the matrix is infinite, add a further column with \cdots
1248     \bool_if:NT \l__physicx_matrix_infinite_bool
1249     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1250 }
1251 % if the matrix is infinite, add a final row
1252 \bool_if:NT \l__physicx_matrix_infinite_bool
1253 {
1254     % finish up the row
1255     \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1256     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1257     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1258     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1259     \bool_if:NT \l__physicx_matrix_dotcol_bool
1260     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1261     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1262     % update cols
1263     \bool_if:NTF \l__physicx_matrix_dotcol_bool
1264     { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1265     { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1266 }
1267 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1268 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1269 {
1270   \keyval_parse:nnn
1271   \__physicx_matrix_diag_parse_aux:n
1272   \__physicx_matrix_diag_parse_aux:nn
1273   {#1}
1274 }
1275 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1276 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1277 {
1278   \str_case_e:nnF {#1}
1279   {
1280     { auto-update }
1281     {
1282       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1283       \__physicx_matrix_calc:nn
1284     }
1285     { noauto-update }
1286     {
1287       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1288     }
1289     { true }
1290     {
1291       \bool_set_true:N \l__physicx_matrix_diag_bool
1292       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1293       \__physicx_diagonalmatrix_set_diag:
1294     }
1295     { false }
1296     {
1297       \bool_set_false:N \l__physicx_matrix_diag_bool
1298       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1299       \__physicx_diagonalmatrix_no_diag:
1300     }
1301   }
1302   { \msg_error:nnn { physicx } { diag-key } {#1} }
1303 }
1304 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1305 {
1306   \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
1307   \tl_set:Nx \l__physicx_tmpdiag_tl
1308   { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1309   \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
1310   \tl_if_head_eq_charcode:nNTF {#1} '
1311   {
1312     \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1313     { \tl_tail:n {#1} }
1314   }
1315   { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1316 }
1317 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1318 {

```

```

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

```

```

1373         {##1}
1374         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1375         {##2}
1376     }
1377 \else:
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 + #1 }
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 + 1 } }
1387         {##2}
1388     }
1389 \else:
1390     \__physicx_matrix_diag_calc:nn
1391     { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1392     { \seq_count:N \l__physicx_tmpdiag_seq }
1393     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1394     {
1395         \physicx_matrix_set_r_c:nnn
1396         { \int_eval:n { ##1 - #1 } }
1397         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1398         {##2}
1399     }
1400 \fi:
1401 \fi:
1402 }
1403 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1404 { \__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
1405 \cs_new:Npn \physicx_matrix_item_parse:n #1
1406 {
1407     \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1408     \keyval_parse:NNn
1409     \__physicx_matrix_item_parse_aux:n
1410     \__physicx_matrix_item_parse_aux:nn
1411     {#1}
1412 }
1413 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1414 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1415 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1416 {
1417     \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1418     \tl_set:Nx \l__physicx_tmpitem_tl
1419     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1420     \physicx_parse_range:neN \l__physicx_matrix_rows_int
1421     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1422     \physicx_parse_range:neN \l__physicx_matrix_cols_int

```

```

1423     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1424 \exp_args:No \tl_if_eq:nnTF
1425 { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1426 {
1427     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1428     {
1429         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1430         {
1431             \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1432         }
1433     }
1434 }
1435 {
1436     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1437     {
1438         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1439         {
1440             \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1441             {
1442                 \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1443                 {##1} {####1} { \l__physicx_tmpitem_tl }
1444             }
1445         }
1446     }
1447 }
1448 }

```

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

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

```

\physicx_matrix_array_parse:o
1449 \cs_new:Npn \physicx_matrix_array_parse:n #1
1450 {
1451     \tl_set:Nn \l__physicx_tmparr_tl {#1}
1452     \tl_set:Nx \l__physicx_tmparr_tl
1453     { \__physicx_expand:w \l__physicx_tmparr_tl }
1454     \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1455     \__physicx_matrix_autocalc:nn
1456     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1457     { 0 }
1458     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1459     {
1460         \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1461         \__physicx_matrix_autocalc:nn
1462         { 0 }
1463         { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1464         \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1465         {
1466             \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1467         }
1468     }
1469 }
1470 \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.

```

1471 \cs_new:Npn \physicx_matrix_array_parse_main:
1472 {
1473   \int_step_inline:nn \l__physicx_matrix_rows_int
1474   {
1475     \int_step_inline:nn \l__physicx_matrix_cols_int
1476     {
1477       \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1478       {##1} {####1} \l__physicx_matrix_main_tl
1479     }
1480   }
1481 }

```

(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).

```

1482 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1483 {
1484   \physicx_if_num:nTF {#1}
1485   { \prg_return_true: }
1486   {
1487     \bool_case_true:nTF
1488     {
1489       { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1490       { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1491       {
1492         \bool_lazy_and_p:nn
1493         { \cs_if_exist_p:N \inteval }
1494         { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1495       } { }
1496       {
1497         \bool_lazy_and_p:nn
1498         { \cs_if_exist_p:N \fpeval }
1499         { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1500       } { }
1501     }
1502     { \prg_return_true: }
1503     { \prg_return_false: }
1504   }
1505 }

```

(End definition for __physicx_if_can_num:n.)

\diagonalmatrix Define \diagonalmatrix.

```

1506 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{ } m }
1507 {
1508   \group_begin:
1509   \IfBooleanTF {#1}
1510   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicx_tmp } }
1511   { \keys_set:nn { physicx/matrix } { #3 } }
1512   \physicx_construct:nnn { }
1513   {
1514     \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist

```

```

1515     \tl_if_empty:nF {#4}
1516     {
1517         \__physicx_if_keyval:nTF {#4}
1518         { \physicx_matrix_diag_parse:n { true, #4 } }
1519         { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1520     }
1521 }
1522 { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1523 \bool_lazy_or:nnTF
1524 { \bool_if_p:n {#2} }
1525 { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1526 {
1527     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1528     {
1529         \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1530         \__physicx_matrix_appto_body_e:off
1531     }
1532     {
1533         \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1534         \__physicx_matrix_appto_body_ne:off
1535     }
1536     \use_i_ii:nnn
1537 }
1538 { \use_i:nn }
1539 \__physicx_matrix_transpose:N
1540 \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1541 \__physicx_diagonalmatrix_generate_body:NNN
1542 \__physicx_matrix_save_or_print:
1543 \group_end:
1544 }
1545 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1546 {
1547     \__physicx_matrix_generate_body:NNNN #1#2#3
1548     \__physicx_diagonalmatrix_enhanced:nnn
1549 }
1550 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1551 {
1552     \int_step_inline:nn { #1 - 1 }
1553     {
1554         \int_step_inline:nn { #2 - 1 }
1555         {
1556             \tl_put_right:Nx \l__physicx_matrix_body_tl
1557             {
1558                 \exp_after:wN
1559                 \physicx_matrix_use_r_c:nn
1560                 #3 {{##1}} {{####1}} &
1561             }
1562         }
1563     }
1564     \tl_put_right:Nx \l__physicx_matrix_body_tl
1565     {
1566         \exp_after:wN
1567         \physicx_matrix_use_r_c:nn
1568         #3 {{##1}} {{ \int_use:N #2 }} \__physicx_matrix_sep:
1569     }

```

```

1569     }
1570     \int_step_inline:nn { #2 - 1 }
1571     {
1572         \tl_put_right:Nx \l__physicx_matrix_body_tl
1573         {
1574             \exp_after:wN
1575             \physicx_matrix_use_r_c:nn
1576             #3 {{ \int_use:N #1 }} {{##1}} &
1577         }
1578     }
1579     \tl_put_right:Nx \l__physicx_matrix_body_tl
1580     {
1581         \exp_after:wN
1582         \physicx_matrix_use_r_c:nn
1583         #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1584     }
1585 }

```

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

__physicx_declare_init:

```

1586 \cs_new:Npn \__physicx_matrix_enhanced_init:
1587 {
1588     \seq_if_empty:NF \l__physicx_row_list_seq
1589     {
1590         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1591         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1592         { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1593     }
1594     \seq_if_empty:NF \l__physicx_col_list_seq
1595     {
1596         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1597         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1598         { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1599     }
1600 }

```

(End definition for __physicx_declare_init:.)

\commamatrix Define \commamatrix.

```

1601 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1602 {
1603     \group_begin:
1604     \keys_set:nn { physicx/matrix } {#3}
1605     \tl_if_empty:nF {#4}
1606     { \keys_set:nn { physicx/matrix } { array = {#4} } }
1607     \IfBooleanT {#1}
1608     { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }
1609     \tl_set:Nx \l__physicx_matrix_array_tl
1610     { \__physicx_expand:w \l__physicx_matrix_array_tl }
1611     \bool_lazy_or:nnTF
1612     { \bool_if_p:n {#2} }
1613     { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1614     { \__physicx_commamatrix_enhanced: }
1615     {

```



```

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

```

```

1670 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1671 {
1672   \seq_set_split:Non \l__physicx_tmp_col_seq
1673   { \physicx@align } {#3}
1674   \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1675   \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1676   { #2 {##2} {#1} {##1} }
1677   \tl_put_right:Nx \l__physicx_matrix_body_tl
1678   {
1679     \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1680     \if_int_compare:w \l__physicx_matrix_rows_int = #1
1681       \scan_stop:
1682     \else:
1683       \__physicx_matrix_sep:
1684     \fi:
1685   }
1686 }
1687 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1688 {
1689   \seq_put_right:Nx \l__physicx_tmp_coled_seq
1690   {
1691     \text_expand:n % \text_expand:n do the magic thing, but slower
1692     {
1693       \physicx@matricelement { #1 }
1694       { \__physicx_matrix_row_iterate:n {#2} }
1695       { \__physicx_matrix_col_iterate:n {#3} }
1696     }
1697   }
1698 }
1699 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1700 {
1701   \seq_put_right:No \l__physicx_tmp_coled_seq
1702   {
1703     \physicx@matricelement {#1}
1704     { \__physicx_matrix_row_iterate:n {#2} }
1705     { \__physicx_matrix_col_iterate:n {#3} }
1706   }
1707 }

```

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

\generalmatrix Define \generalmatrix.

```

1708 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1709 {
1710   \IfBooleanTF {#2}
1711   {
1712     \group_begin:
1713     \IfBooleanTF {#1}
1714     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1715     { \keys_set:nn { physicx/matrix } {#4} }
1716     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1717     \physicx_construct:nnn
1718     {
1719       \tl_if_empty:NTF \l__physicx_matrix_main_tl

```

```

1720         {
1721             \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1722         }
1723         { \physicx_matrix_array_parse_main: }
1724     }
1725     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1726     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1727     \__physicx_generalmatrix:
1728     \__physicx_matrix_save_or_print:
1729     \group_end:
1730 }
1731 {
1732     \IfBooleanTF {#1}
1733     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1734     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1735     \qxmatrix = * [#4]
1736 }
1737 }
1738 \cs_new:Npn \__physicx_generalmatrix:
1739 {
1740     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1741     {
1742         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1743         \__physicx_matrix_appto_body_e:off
1744     }
1745     {
1746         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1747         \__physicx_matrix_appto_body_ne:off
1748     }
1749     \__physicx_matrix_transpose:N
1750     \__physicx_matrix_generate_body:NNNN
1751     \__physicx_generalmatrix_generate:nnn
1752 }

```

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

__physicx_matrix_generate_body:NNNN

```

1753 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1754 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1755 {
1756     \__physicx_matrix_enhanced_init:
1757     \int_step_inline:nn { #1 - 1 }
1758     {
1759         \int_step_inline:nn { #2 - 1 }
1760         {
1761             \tl_set:Nx \l__physicx_tmp_tl
1762             {
1763                 \exp_after:wN
1764                 \physicx_matrix_use_r_c:nn
1765                 #3 {{##1}} {{####1}}
1766             }
1767             #4 \l__physicx_tmp_tl {##1} {####1}
1768             \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1769         }

```

```

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

```

(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
1799 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1800 {
1801   \tl_put_right:Nx \l__physicx_matrix_body_tl
1802   {
1803     \text_expand:n
1804     {
1805       \physicx@matricelement {#1}
1806       { \__physicx_matrix_row_iterate:n {#2} }
1807       { \__physicx_matrix_col_iterate:n {#3} }
1808     }
1809   }
1810 }
1811 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1812 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1813 {
1814   \tl_put_right:No \l__physicx_matrix_body_tl
1815   {
1816     \physicx@matricelement {#1}
1817     { \__physicx_matrix_row_iterate:n {#2} }
1818     { \__physicx_matrix_col_iterate:n {#3} }
1819   }

```



```

1862     {
1863         \physicsx_matrix_set_r_c:nnn
1864         { \int_eval:n { \l__physicsx_matrix_rows_int - 1 } }
1865         { \int_eval:n { \l__physicsx_matrix_cols_int - 1 } }
1866         { \ddots }
1867     }
1868     \bool_if:NT \l__physicsx_matrix_infinite_bool
1869     {
1870         \int_incr:N \l__physicsx_matrix_rows_int
1871         \int_incr:N \l__physicsx_matrix_cols_int
1872         \__physicsx_matrix_last_aux_c:
1873         \__physicsx_matrix_last_aux_r:
1874         \physicsx_matrix_set_r_c:nnn
1875         { \int_use:N \l__physicsx_matrix_rows_int }
1876         { \int_use:N \l__physicsx_matrix_cols_int }
1877         { \ddots }
1878     }
1879     \l__physicsx_matrix_ending_tl
1880 }
1881 \cs_new:Npn \__physicsx_matrix_last_aux_c:
1882 {
1883     \int_step_inline:nn \l__physicsx_matrix_rows_int
1884     {
1885         \physicsx_matrix_set_r_c:nnn
1886         {##1} { \int_use:N \l__physicsx_matrix_cols_int }
1887         { \cdots }
1888     }
1889 }
1890 \cs_new:Npn \__physicsx_matrix_last_aux_r:
1891 {
1892     \int_step_inline:nn \l__physicsx_matrix_cols_int
1893     {
1894         \physicsx_matrix_set_r_c:nnn
1895         { \int_use:N \l__physicsx_matrix_rows_int } {##1}
1896         { \vdots }
1897     }
1898 }

```

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

1.3.3 Define new matrix command

```

\__physicsx_new_matrix_cmd:NNN
\newgeneralmatrix 1899 \cs_new:Npn \__physicsx_new_matrix_cmd:NNN #1#2#3
\NewGeneralMatrix 1900 {
\newdiagonalmatrix 1901     \NewDocumentCommand #2 { t+ m o o m m }
\NewDiagonalMatrix 1902     {
\newcommamatrix 1903         \IfBooleanTF {##1}
\NewCommaMatrix 1904         {
1905             \IfNoValueTF {##3}
1906             { \newcommand ##2 { #1 + [##5] {##6} } }
1907             {
1908                 \IfNoValueTF {##4}
1909                 { \newcommand ##2 [##3] { #1 + [##5] {##6} } }

```

```

1910         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1911     }
1912 }
1913 {
1914     \IfNoValueTF {##3}
1915     { \newcommand ##2 { #1 [##5] {##6} } }
1916     {
1917         \IfNoValueTF {##4}
1918         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1919         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1920     }
1921 }
1922 }
1923 \NewDocumentCommand #3 { t+ m m m m }
1924 {
1925     \IfBooleanTF {##1}
1926     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1927     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1928 }
1929 }
1930 \_physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1931 \_physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1932 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1933 {
1934     \IfBooleanTF {#1}
1935     {
1936         \IfNoValueTF {#3}
1937         { \newcommand #2 { \generalmatrix + {#5} } }
1938         {
1939             \IfNoValueTF {#4}
1940             { \newcommand #2 [3] { \generalmatrix + {#5} } }
1941             { \newcommand #2 [3] [4] { \generalmatrix + {#5} } }
1942         }
1943     }
1944     {
1945         \IfNoValueTF {#3}
1946         { \newcommand #2 { \generalmatrix {#5} } }
1947         {
1948             \IfNoValueTF {#4}
1949             { \newcommand #2 [3] { \generalmatrix {#5} } }
1950             { \newcommand #2 [3] [4] { \generalmatrix {#5} } }
1951         }
1952     }
1953 }
1954 \NewDocumentCommand \NewGeneralMatrix { t+ m m m m }
1955 {
1956     \IfBooleanTF {#1}
1957     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1958     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1959 }

```

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

```

1960 \end{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	
\+	177
\-	177
\.	177
\\	57, 1841
\{	428, 666, 685, 703, 744, 769
\}	428, 666, 687, 703, 744, 769
_	177
A	
\A	177
\abs	<u>737</u>
\absolutevalue	<u>737</u>
\acommm	<u>737</u>
\aftergroup	227
\anticommutator	<u>737</u>
\AtBeginDocument	272, 295, 308, 319, 332
B	
\begin	433, 434, 435, 436, 437, 438, 439, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 686, 689, 693, 696, 697, 698, 703, 704, 708, 711, 712, 713, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099
\bgroup	226, 228
\Big	719
\big	719
\Bigg	719
\bigg	719
\Biggl	726
\biggl	725
\Biggr	726
\biggr	725
\Bigl	724
\bigl	723
\Bigr	724
\bigr	723
\boldsymbol	216
bool commands:	
\bool_case_false:n	636, 637, 644
\bool_case_true:nTF	721, 1487
\bool_if:N	16, 21, 26, 85, 205, 270, 311, 317, 330, 650, 652, 656, 1149, 1204, 1211, 1221, 1229, 1241, 1248, 1252, 1259, 1263, 1527, 1652, 1740, 1824, 1868
\bool_if_p:N	1525, 1613
\bool_if_p:n	723, 724, 725, 726, 1524, 1612
\bool_lazy_and_p:nn	1492, 1497
\bool_lazy_or:nnTF	31, 137, 152, 481, 486, 1523, 1611, 1859
\bool_new:N	9, 10, 11, 12, 13, 51, 66, 552, 554, 828, 829, 830, 838, 852, 860, 861
\bool_set:Nn	563, 564, 1716
\bool_set_false:N	95, 193, 282, 1137, 1165, 1180, 1297
\bool_set_inverse:N	898, 1024
\bool_set_true:N	140, 155, 165, 200, 268, 281, 1136, 1169, 1173, 1184, 1188, 1291, 1590, 1596
\c_false_bool	681, 701
\c_true_bool	664, 674, 681, 701
\Bqty	737
\bqty	<u>737</u>
C	
\cdots	924, 1203, 1206, 1207, 1210, 1212, 1243, 1244, 1247, 1249, 1887
char commands:	
\char_value_catcode:n	396
clist commands:	
\clist_clear:N	1105
\clist_concat:NNN	1112
\clist_if_empty:nTF	242
\clist_if_in:NnTF	1440
\clist_map_break:n	186, 198
\clist_map_inline:nn	82, 182, 194, 1110
\clist_new:N	50, 836, 837
\clist_put_right:Nn	986, 992, 1431
\clist_set:Nn	1102
\clist_set_eq:NN	1407
\c_empty_clist	1407
\comm	767
\commamatrix	32, <u>1601</u> , 1931
\commutator	<u>737</u>
\crossproduct	304
cs commands:	
\cs:w	32, 33, 1015
\cs_end:	32, 33, 791, 1015

fp commands:		<code>\int_use:N</code> 1567, 1576, 1583, 1774, 1776, 1786, 1788, 1795, 1797, 1875, 1876, 1886, 1895
<code>\fp_compare:nNnTF</code> 236		<code>\int_zero:N</code> 1319, 1646
<code>\fp_eval:n</code> 30, 238, 389, 1490		<code>\c_zero_int</code> 395
<code>\fpeval</code> 30, 1498, 1499		<code>\inteval</code> 30, 1493, 1494
G		
<code>\generalmatrix</code> . . 34, 1708, 1937, 1940, 1941, 1946, 1949, 1950, 1957, 1958		
group commands:		K
<code>\group_begin:</code>		keys commands:
. 468, 1134, 1508, 1603, 1712		<code>\keys_define:nn</code>
<code>\group_end:</code> 476, 1145, 1543, 1624, 1729	 211, 245, 360, 362, 866, 868
H		
<code>\hat</code> 301		<code>\keys_if_exist:nnTF</code> 184, 196
hook commands:		<code>\l_keys_key_str</code> 184, 187, 196, 201, 396, 399, 402, 408, 410, 1053, 1056
<code>\hook_gput_code:nnn</code> 37		<code>\l_keys_path_str</code> 420, 1060
I		
if commands:		<code>\keys_set:nn</code> 3, 187, 201, 349, 350, 361, 401, 410, 460, 469, 867, 1055, 1118, 1141, 1142, 1510, 1511, 1604, 1606, 1608, 1714, 1715
<code>\if_bool:N</code> 597, 609, 662, 737		keyval commands:
<code>\if_case:w</code> 589		<code>\keyval_parse:NNn</code> 1408
<code>\if_cs_exist:w</code> 791		<code>\keyval_parse:nnn</code> 1270
<code>\if_int_compare:w</code> 582, 1213, 1340, 1343, 1366, 1378, 1635, 1680		
<code>\IfBooleanT</code> 685, 687, 1607		L
<code>\IfBooleanTF</code> . . 297, 299, 301, 348, 544, 635, 692, 694, 707, 709, 729, 1070, 1135, 1140, 1509, 1710, 1713, 1732, 1733, 1734, 1903, 1925, 1934, 1956		<code>\langle</code> 432, 670, 697, 712
<code>\IfNoValueTF</code>		<code>\lbrace</code> 312, 313
. . 312, 313, 511, 517, 1905, 1908, 1914, 1917, 1936, 1939, 1945, 1948		<code>\ldots</code> 924
<code>\Im</code> 276, 313, 323		<code>\left</code> 226, 371, 462, 487, 685, 692, 697, 703, 704, 707, 711, 712, 713
<code>\imaginary</code> 276, 313, 323		<code>\let</code> 312, 313, 630
int commands:		<code>\lgroup</code> 692, 707
<code>\int_compare:nNnTF</code> . 148, 150, 164, 395		<code>\lVert</code> 431, 754
<code>\int_compare:nTF</code> 1163, 1178		<code>\lvert</code> 430
<code>\int_compare_p:nNn</code> . 138, 139, 153, 154		
<code>\int_eval:n</code> . . 30, 1347, 1356, 1374, 1386, 1396, 1397, 1489, 1864, 1865		M
<code>\int_incr:N</code> . 581, 1322, 1656, 1664, 1849, 1851, 1855, 1857, 1870, 1871		<code>\mathbf</code> 297, 299, 301
<code>\int_max:nn</code> 784, 786		<code>\mathcal</code> 212, 213
<code>\int_new:N</code>		<code>\mathclose</code> 226, 228
. . 52, 53, 62, 63, 64, 65, 556, 833, 834		<code>\mathopen</code> 226, 228
<code>\int_set:Nn</code> . . 80, 81, 103, 106, 562, 783, 785, 1138, 1139, 1166, 1181, 1649		<code>\matrixquantity</code> 662
<code>\int_set_eq:NN</code>		<code>\mqty</code> 737
. 102, 105, 149, 151, 1325, 1636		msg commands:
<code>\int_step_inline:nn</code>		<code>\msg_error:nnn</code> 1302
. . . 1191, 1473, 1475, 1552, 1554, 1570, 1757, 1759, 1780, 1883, 1892		<code>\msg_error:nnnn</code> 419, 1059
<code>\int_step_inline:nnn</code>		<code>\msg_new:nnn</code> 60
. 157, 167, 1197, 1235		<code>\msg_new:nnnn</code> 54
		N
		<code>\nabla</code> 294
		<code>\NewCommaMatrix</code> 1899
		<code>\newcommamatrix</code> 1899
		<code>\newcommand</code> 514, 520, 524, 533, 1906, 1909, 1910, 1915, 1918, 1919, 1937, 1940, 1941, 1946, 1949, 1950
		<code>\NewDiagonalMatrix</code> 1899

<code>\newdiagonalmatrix</code>	1899	<code>\physicx_matrix_item_parse:n</code>	993 , 1405 , 1405 , 1413 , 1522 , 1726
<code>\NewDocumentCommand</code> ...	346 , 425 , 507 , 509 , 540 , 542 , 1068 , 1901 , 1923 , 1926 , 1927 , 1932 , 1954 , 1957 , 1958	<code>\physicx_matrix_new_type:nn</code>	1064 , 1066 , 1071
<code>\NewGeneralMatrix</code>	1899	<code>\physicx_matrix_new_type:nnn</code>	1064 , 1064 , 1072
<code>\newgeneralmatrix</code>	1899	<code>\physicx_matrix_set_r_c:nnn</code>	822 , 965 , 968 , 971 , 974 , 977 , 1323 , 1331 , 1346 , 1355 , 1372 , 1384 , 1395 , 1442 , 1466 , 1477 , 1863 , 1874 , 1885 , 1894
<code>\NewXQuantity</code>	466	<code>\physicx_matrix_use_r_c:nn</code>	789 , 1559 , 1566 , 1575 , 1582 , 1764 , 1773 , 1785 , 1794
<code>\newxquantity</code>	466	<code>\physicx_new_type:nnn</code>	210 , 426 , 1065 , 1067
<code>\nobreak</code>	676 , 677 , 678	<code>\physicx_option_or:nn</code>	29
<code>\norm</code>	737	<code>\physicx_option_or:nnTF</code>	738 , 748 , 766 , 770
O				
<code>\OOrder</code>	777	<code>\c_physicx_Order_tl</code>	213 , 287 , 758 , 778
<code>\oorder</code>	737	<code>\c_physicx_order_tl</code>	212 , 286 , 762 , 780
<code>\operatorname</code>	312 , 313	<code>\physicx_parse_range:nnN</code>	90 , 92 , 1420 , 1422
or commands:		<code>\physicx_parse_range:nnnN</code>	62 , 77 , 89 , 91
<code>\or:</code>	590 , 591 , 592	<code>\physicx_parse_range_check:</code>	62 , 67 , 997
<code>\order</code>	737	<code>\physicx_parse_range_nocheck:</code>	62 , 72 , 998
<code>\Ordersymbol</code>	287	<code>\physicx_qxmatrix:nnn</code>	1143 , 1147
<code>\ordersymbol</code>	286	<code>\physicx_right:</code>	227 , 233 , 489 , 631 , 731
P				
<code>\PassOptionsToPackage</code>	242	<code>\physicx_right:N</code>	229 , 501 , 723 , 724 , 725 , 726
<code>\pb</code>	737	<code>\physicx_right:nN</code>	232
peek commands:		<code>\physicx_search_also:nn</code>	180 , 191 , 208 , 209
<code>\peek_charcode_ignore_spaces:NTF</code>	630	<code>\physicx_search_also:nnTF</code>	413 , 1045
physicx commands:		<code>\physicx_set_parse_range_-</code>	110 , 134
<code>\physicx_bf:</code>	216 , 220 , 224 , 297 , 299 , 301	<code>delimiter:n</code>	19
<code>\physicx_compat:</code>	14	<code>\physicx_short:</code>	44
<code>\physicx_compat:TF</code>	284 , 755	<code>\physicx_unimath:</code>	40 , 41 , 42 , 293
<code>\physicx_construct:nnn</code> 1512 , 1717 , 1843 , 1843	<code>\physicx_use_amssymb_type:</code>	214 , 292
<code>\physicx_declare_legacy_paren:NnnnNnN</code> 716 , 717 , 736 , 741 , 742 , 743 , 744 , 746 , 753 , 757 , 761 , 764 , 768 , 777 , 779	<code>\physicx_use_uni_bf_type:</code>	222
<code>\physicx_declare_legacy_quantity:nnNn</code> 550 , 567 , 661 , 663 , 673 , 680 , 700	<code>\physicx_use_uni_bfit_type:</code>	218 , 291
<code>\physicx_if_num:n</code>	175	<code>\physicx_xquantity:nn</code>	466 , 466 , 507 , 529 , 532 , 545 , 546
<code>\physicx_if_num:nTF</code> 386 , 408 , 1053 , 1161 , 1176 , 1484	physicx internal commands:		
<code>\physicx_left:</code>	226 , 231 , 489 , 631 , 731	<code>__physicx_adi:nnn</code>	863 , 1029 , 1030 , 1031 , 1032 , 1033 , 1034 , 1846
<code>\physicx_left:N</code> 228 , 499 , 723 , 724 , 725 , 726	<code>\l__physicx_begin_range_int</code>	62 , 102 , 103 , 148 , 149 , 153 , 154 , 158 , 164 , 168
<code>\physicx_left:nN</code>	230 , 492	<code>\l__physicx_cmd_arg_int</code>	556 , 562 , 581 , 582 , 589
<code>\physicx_mathtools:</code>	24			
<code>\physicx_mathtools:TF</code>	440 , 1081			
<code>\physicx_matrix_array_parse:n</code> 1449 , 1449 , 1470 , 1721			
<code>\physicx_matrix_array_parse_-</code>			
<code>main:</code>	1471 , 1471 , 1723			
<code>\physicx_matrix_diag_parse:n</code> 987 , 1268 , 1268 , 1275 , 1514 , 1518 , 1519 , 1725			

\l__physicx_cmd_arg_spec_tl	__physicx_if_keyval:nTF . . 173, 1517
..... 555, 561, 574, 583	\l__physicx_invalid_range_bool . .
\l__physicx_cmd_auto_body_bool 66, 85, 95, 140, 155, 165
..... 554, 564, 609, 652, 656	\l__physicx_item_ignore_clist . . .
\l__physicx_cmd_auto_body_tl 1407, 1431, 1440
..... 553, 560, 576, 610, 611	__physicx_loadpackage_options:nnn
\l__physicx_cmd_noauto_body_bool 240, 251, 253, 255
..... 552, 563, 597, 650	\g__physicx_mathtools_bool
\l__physicx_cmd_noauto_body_tl 9, 26, 281, 282
..... 551, 559, 575, 598, 599	__physicx_matrix_add_data:nn . . .
\l__physicx_col_list_seq 989, 995, 1106, 1108
..... 840, 894, 1594, 1598	\l__physicx_matrix_after_begin_-
__physicx_commamatrix_enhanced:	tl 858, 1005, 1007, 1638
..... 1614, 1643	\l__physicx_matrix_after_end_tl . .
__physicx_commamatrix_enhanced_- 859, 1008, 1010, 1640
aux:nNn 1657, 1665, 1670	__physicx_matrix_appto_body_-
__physicx_commamatrix_enhanced_-	e:nnn
aux_e:nnn 1658, 1687	.. 1152, 1530, 1743, 1799, 1799, 1811
__physicx_commamatrix_enhanced_-	__physicx_matrix_appto_body_-
aux_ne:nnn 1666, 1699	ne:nnn
\g__physicx_compat_bool 1156, 1534, 1747, 1799, 1812, 1821
..... 11, 16, 247, 268, 270	\l__physicx_matrix_args_tl
__physicx_declare_init: 1586 857, 1003, 1004, 1638
__physicx_declare_init:nnn 557, 569	\l__physicx_matrix_array_tl
__physicx_declare_legacy_- 831, 870, 1609,
quantity_aux:nnnn 579, 620	1610, 1616, 1618, 1621, 1648, 1721
__physicx_declare_legacy_-	__physicx_matrix_autocalc:nn . . .
quantity_aux:NNnnn . . 572, 623, 647 851, 886, 888, 1404, 1455, 1461
__physicx_declare_legacy_-	__physicx_matrix_begin:w
quantity_aux:nw 570, 616, 621 849, 1000, 1638
__physicx_delsize:NNnn 231, 233, 234	\l__physicx_matrix_beginning_tl . .
__physicx_diagonalmatrix_diag_- 864, 1035, 1037, 1845
main: 1292, 1298, 1336, 1341	\l__physicx_matrix_body_tl
__physicx_diagonalmatrix_- 832, 1159, 1194,
enhanced:nnn 1529, 1533, 1548	1199, 1200, 1206, 1207, 1212, 1217,
__physicx_diagonalmatrix_-	1224, 1226, 1228, 1230, 1231, 1233,
generate_body:NNN 1541, 1550	1237, 1238, 1243, 1244, 1249, 1255,
__physicx_diagonalmatrix_-	1256, 1258, 1260, 1261, 1556, 1563,
generate_enhanced_body:NNN . .	1572, 1579, 1620, 1632, 1639, 1645,
..... 1540, 1545	1677, 1768, 1777, 1789, 1801, 1814
__physicx_diagonalmatrix_no_-	__physicx_matrix_calc:nn
diag: 1299, 1328, 1337 781, 886, 1283
__physicx_diagonalmatrix_set_-	__physicx_matrix_col_iterate:n . .
diag: 1293, 1317	848, 981, 1597, 1695, 1705, 1807, 1818
__physicx_do_nothing: 115	\l__physicx_matrix_cols_int
\l__physicx_end_range_int . . . 63, 785, 786,
105, 106, 150, 151, 154, 158, 164, 168	834, 883, 1139, 1178, 1181, 1197,
__physicx_expand:w 842, 873,	1227, 1235, 1257, 1264, 1265, 1319,
875, 877, 880, 1308, 1419, 1453, 1610	1322, 1326, 1374, 1386, 1397, 1422,
\g__physicx_fixdif_bool 261, 311, 330	1475, 1635, 1636, 1827, 1834, 1849,
__physicx_generalmatrix: 1727, 1738	1851, 1865, 1871, 1876, 1886, 1892
__physicx_generalmatrix_-	\l__physicx_matrix_diag_bool . . .
generate:nnn 1742, 1746, 1751 838, 1291, 1297
__physicx_if_can_num:n . . 1482, 1482	

```

\__physicx_matrix_diag_calc:nn . .
    . . . . . 1282, 1287, 1332,
    1349, 1358, 1367, 1379, 1390, 1403
\l_physicx_matrix_diag_clist . .
    . . . . . 836, 984, 986, 1514, 1725
\__physicx_matrix_diag_parse_-
    aux:n . . . . . 1271, 1276
\__physicx_matrix_diag_parse_-
    aux:nn . . . . . 1272, 1304
\__physicx_matrix_diag_parse_-
    aux_anti:n . . . . . 1312, 1364
\__physicx_matrix_diag_parse_-
    aux_regu:n . . . . . 1315, 1338
\l_physicx_matrix_dotcol_bool . .
    . . . . . 830, 1180, 1184,
    1188, 1204, 1229, 1241, 1259, 1263
\l__physicx_matrix_dotrow_bool . .
    . . . . . 829, 1165, 1169, 1173, 1221
\__physicx_matrix_element_-
    aux:nnn . . . . 902, 908, 911, 917,
    920, 926, 929, 935, 938, 944, 947, 952
\__physicx_matrix_end: . . . 1001, 1640
\__physicx_matrix_end:w . . . . . 850
\l_physicx_matrix_ending_tl . .
    . . . . . 865, 1038, 1040, 1879
\l__physicx_matrix_enhanced_bool
    . . . . . 861, 1021, 1024, 1525, 1613
\__physicx_matrix_enhanced_init:
    . . . . . 1586, 1651, 1756
\l_physicx_matrix_expand_-
    element_bool . . . . . 852,
    960, 1149, 1527, 1590, 1596, 1652, 1740
\__physicx_matrix_generate_-
    body:NNNN . . 1547, 1750, 1753, 1754
\l__physicx_matrix_infinite_bool
    . . . . . 828, 895, 898, 1136,
    1137, 1211, 1248, 1252, 1716, 1868
\l_physicx_matrix_item_clist . .
    . . . . . 837, 990, 992, 1522, 1726
\__physicx_matrix_item_parse_-
    aux:n . . . . . 1409, 1414
\__physicx_matrix_item_parse_-
    aux:nn . . . . . 1410, 1415
\__physicx_matrix_last_aux_c: . .
    . . . . . 1850, 1872, 1881
\__physicx_matrix_last_aux_r: . .
    . . . . . 1856, 1873, 1890
\l__physicx_matrix_last_col_tl . .
    . . . . . 847, 983, 1847, 1861
\l__physicx_matrix_last_row_tl . .
    . . . . . 846, 982, 1853, 1860
\l__physicx_matrix_main_tl . . . .
    . . . . . 835, 890, 1478, 1719

\__physicx_matrix_row_iterate:n .
    845, 980, 1591, 1694, 1704, 1806, 1817
\l__physicx_matrix_rows_int . . .
    . . . . . 783, 784, 833, 882, 1138,
    1163, 1166, 1191, 1213, 1325, 1420,
    1473, 1649, 1680, 1828, 1833, 1855,
    1857, 1864, 1870, 1875, 1883, 1895
\__physicx_matrix_save_or_print:
    . . . . . 1144, 1542, 1623, 1626, 1728
\l__physicx_matrix_save_tl . . . .
    . . . . . 1013, 1015, 1628, 1631
\__physicx_matrix_sep: . . . . .
    . . . . . 1217, 1224, 1231, 1255,
    1567, 1617, 1683, 1778, 1838, 1838
\l__physicx_matrix_sep_dim . . . .
    . . . . . 862, 1011, 1840, 1841
\__physicx_matrix_set_data:nn . .
    . . . . . 988, 994, 1103
\__physicx_matrix_set_r_c_-
    ckall:nnn . . . . . 820, 978
\__physicx_matrix_set_r_c_-
    ckep:nnn . . . . . 805, 969
\__physicx_matrix_set_r_c_-
    ckig:nnn . . . . . 800, 972
\__physicx_matrix_set_r_c_-
    ckigep:nnn . . . . . 811, 821, 975
\__physicx_matrix_set_r_c_-
    nock:nnn . . . . . 798, 823, 966
\l__physicx_matrix_tmparr_c_sep .
    . . . . . 1460, 1463, 1464
\l__physicx_matrix_tmparr_r_sep .
    . . . . . 1454, 1456, 1458
\__physicx_matrix_transpose:N . .
    . . . . . 1539, 1749, 1822, 1822
\l__physicx_matrix_transpose_-
    bool . . . . . 860, 1016, 1824
\l__physicx_max_range_int . . . .
    . . . . 64, 81, 105, 138, 150, 151, 153
\l__physicx_min_range_int . . . .
    . . . . . 65, 80, 102, 139, 148, 149
\__physicx_nauto_case:nnnn . 625, 648
\__physicx_new_matrix_cmd:NNN . .
    . . . . . 1899, 1899, 1930, 1931
\__physicx_new_xquantity_aux:w . .
    . . . . . 513, 519, 523, 527, 530
\g__physicx_original_bool . . 262, 317
\__physicx_parse_range_action:nnn
    . . . . . 97, 114, 124
\__physicx_parse_range_aux:n 84, 93
\__physicx_parse_range_aux:w . .
    . . . . . 115, 116, 125, 126
\__physicx_parse_range_range: . .
    . . . . . 70, 75, 107, 172

```

__physicx_parse_range_range_-	\l__physicx_tmpa_int
check: 70, 146, 172 52, 1646, 1656, 1658, 1664, 1666
__physicx_parse_range_range_-	\l__physicx_tmpa_seq
nocheck: 75, 162 86, 96, 141, 144, 159, 169
__physicx_parse_range_single:n .	\l__physicx_tmpa_tl
..... 69, 74, 99, 145 101, 103, 118, 119, 128
__physicx_parse_range_single_-	\l__physicx_tmparr_tl
check:n 69, 135, 145 1451, 1452, 1453, 1454
__physicx_parse_range_single_-	\l__physicx_tmpb_int
nocheck:n 74, 143 53
\g__physicx_physics_bool 10	\l__physicx_tmpb_tl ... 104, 106, 129
\l__physicx_quantity_args_tl ...	\l__physicx_tmpdiag_seq 1309, 1320,
..... 352, 375, 376, 473	1330, 1333, 1334, 1344, 1350, 1351,
\l__physicx_quantity_code_tl ...	1353, 1359, 1360, 1368, 1369, 1370,
..... 353, 377, 470, 474	1380, 1381, 1382, 1391, 1392, 1393
\l__physicx_quantity_left_size_-	\l__physicx_tmpdiag_tl
tl 354, 482, 487, 492, 494, 499 1306, 1307, 1308, 1309
\l__physicx_quantity_left_tl ...	\l__physicx_tmpitem_tl
..... 355, 366, 472 1417, 1418, 1419, 1425, 1443
\l__physicx_quantity_post_tl ...	__physicx_vnabla:
..... 356, 365, 505 294, 305, 335, 339, 340
\l__physicx_quantity_pre_tl ...	__physicx_xquantity_aux:nnnn ...
..... 357, 364, 480 471, 478, 508
\l__physicx_quantity_right_size_-	\physicxempty . 794, 808, 816, 853, 854, 962
tl 358, 483, 488, 496, 501	\physicxexcept 856, 933, 942, 951, 957, 959
\l__physicx_quantity_right_tl ...	\PHYSICXIGNORE 7, 8, 802, 813, 1425
..... 359, 367, 475	\physicxset 346
__physicx_quantity_size:nn	\physicxtmp ... 550, 1141, 1510, 1608, 1714
..... 368, 369, 384	\poissonbracket 737
__physicx_quantity_unknown:n ...	\pqty 737
..... 382, 393	prg commands:
__physicx_qxmatrix_appto_-	\prg_generate_conditional_-
body:nnn 1151, 1155,	variant:Nnn 209
1195, 1201, 1208, 1234, 1239, 1245	\prg_new_conditional:Npnn
\g__physicx_reqty_bool 14, 19, 24, 29, 175, 191, 1482
..... 13, 257, 662, 737	\prg_replicate:nn 1227, 1257
\l__physicx_row_list_seq	\prg_return_false:
..... 839, 892, 1588, 1592 17, 22, 27, 35, 47, 178, 206, 1503
\g__physicx_short_bool ... 12, 21, 249	\prg_return_true: 17,
\l__physicx_tmp_col_seq .. 1672, 1675	22, 27, 34, 47, 178, 206, 1485, 1502
\l__physicx_tmp_coled_seq	\prg_set_conditional:Npnn 44
..... 1674, 1679, 1689, 1701	\ProcessKeysPackageOptions 265
\l__physicx_tmp_colnum_seq	
..... 1423, 1429, 1438	
\l__physicx_tmp_rownum_seq	
..... 1421, 1427, 1436	
\l__physicx_tmp_seq	
..... 1647, 1650, 1654, 1662	
\l__physicx_tmp_tl	
..... 584, 598, 610, 1761, 1767,	
1770, 1776, 1782, 1788, 1791, 1797	
\l__physicx_tmpa_bool 51, 193, 200, 205	
\l__physicx_tmpa_clist 50	

Q	
\qty 737	
\quantity 662, 740	
quark commands:	
\q_nil 115	
\q_physicx_special . 115, 116, 125, 126	
\quark_if_nil:nTF 120	
\quark_if_recursion_tail_stop:n .	
..... 618, 619	
\q_recursion_stop 571	
\q_recursion_tail 571	
\qxmatrix 1123, 1735	

<code>\qxqty</code>	549	444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458
R		
<code>\rangle</code>	432, 670, 697, 712	
<code>\rbrace</code>	312, 313	
<code>\Re</code>	275, 312, 322	
<code>\real</code>	275, 312, 322	
regex commands:		
<code>\regex_match:nnTF</code>	177, 942, 951	
<code>\renewdif</code>	335, 340	
<code>\RequirePackage</code>	243	
<code>\rgroup</code>	694, 709	
<code>\right</code>	227, 371, 463, 488, 687, 694, 697, 703, 704, 709, 711, 712, 713	
<code>\rVert</code>	431, 754	
<code>\rvert</code>	430	
S		
<code>\sb</code> 844, 1194, 1199, 1206, 1233, 1237, 1243		
scan commands:		
<code>\scan_stop:</code>	1214, 1681	
seq commands:		
<code>\c_empty_seq</code>	79, 1674	
<code>\seq_clear:N</code>	96	
<code>\seq_concat:NNN</code>	86	
<code>\seq_count:N</code>	1333, 1334, 1350, 1351, 1359, 1360, 1368, 1369, 1380, 1381, 1391, 1392, 1456, 1463, 1650	
<code>\seq_if_empty:NTF</code>	1588, 1594	
<code>\seq_item:Nn</code>	1592, 1598	
<code>\seq_map_indexed_inline:Nn</code>	1320, 1330, 1344, 1353, 1370, 1382, 1393, 1458, 1464, 1675	
<code>\seq_map_inline:Nn</code>	1427, 1429, 1436, 1438	
<code>\seq_map_tokens:Nn</code>	1654, 1662	
<code>\seq_new:N</code>	839, 840	
<code>\seq_put_right:Nn</code>	141, 144, 159, 169, 1689, 1701	
<code>\seq_set_eq:NN</code>	79, 1674	
<code>\seq_set_split:Nnn</code>	5, 892, 894, 1309, 1454, 1460, 1647, 1672	
<code>\seq_use:Nn</code>	1679	
<code>\setmathfont</code>	302	
<code>\setmatrixdata</code>	1101	
<code>\setmatrixtype</code>	1042, 1064, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099	
<code>\setquantitytype</code>	380, 425, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 442, 443,	
<code>\smallmatrixquantity</code>	662	
<code>\smqty</code>	737	
str commands:		
<code>\str_case:e:nnTF</code>	1278	
<code>\str_const:Nn</code>	825, 826, 827	
<code>\str_if_eq:nnTF</code>	603, 605	
<code>\symbf</code>	224, 294	
<code>\symbfit</code>	220	
T		
T _E X and L ^A T _E X 2 _ε commands:		
<code>\@declareparencmd</code>	716	
<code>\@declarequantitycmd</code>	550	
<code>\@ifpackageloaded</code>	267, 280, 290, 307, 320, 334	
<code>\@ifstar</code>	535	
<code>\@pkgextension</code>	46	
<code>\bBigg@</code>	238	
<code>\c@MaxMatrixCols</code>	1020, 1635, 1636	
<code>\physics@align</code>	825, 1026, 1460, 1619, 1673	
<code>\physics@cr</code>	826, 1025, 1454, 1617, 1647	
<code>\physics@matrixellement</code>	844, 899, 903, 904, 912, 913, 921, 922, 930, 931, 939, 940, 948, 949, 956, 1693, 1703, 1805, 1816	
<code>\physics@sep</code>	827, 892, 894, 1027, 1309	
tex commands:		
<code>\tex_advance:D</code>	1264, 1265	
<code>\tex_mathclose:D</code>	233	
<code>\tex_mathopen:D</code>	231	
text commands:		
<code>\text_expand:n</code>	875, 1691, 1803	
tl commands:		
<code>\c_empty_tl</code>	372, 642	
<code>\tl_clear:N</code>	559, 560, 561, 1159, 1645	
<code>\tl_const:Nn</code>	212, 213	
<code>\tl_gset_eq:NN</code>	1630	
<code>\tl_head:N</code>	396	
<code>\tl_if_blank:nTF</code>	130, 915	
<code>\tl_if_empty:NTF</code>	101, 104, 1719, 1847, 1853	
<code>\tl_if_empty:nTF</code>	112, 470, 807, 815, 906, 1515, 1605	
<code>\tl_if_empty_p:N</code>	482, 483, 1860, 1861	
<code>\tl_if_eq:nnTF</code>	802, 813, 1424	
<code>\tl_if_exist:NTF</code>	46	
<code>\tl_if_head_eq_charcode:nNTF</code>	1310	
<code>\tl_if_head_eq_meaning:nNTF</code>	491	
<code>\tl_if_head_eq_meaning_p:nN</code>	1489, 1490, 1494, 1499	
<code>\tl_if_in:nnTF</code>	174, 924, 933	

