

The *intexgral* package

v3.0.1

Valentin Dao*

2 January 2026

Abstract

While integral composition is prevalent in \LaTeX , it often proves to be impractical. When the expression becomes complex, it is then difficult to modify its various elements in an illegible source code. To address this problem, the *intexgral* package offers a central macro whose only argument is the integrand. Everything else (symbol, limits, variables...) can be modified using a `\key=value` interface. Unlike the traditional method, where the user chooses the limit order with the active characters `_` and `^`, the package had to establish a convention. Thus, for the two keys that will deal with limits, the assumed input will be `_{lower limit}^{upper limit}`. Since this package is written in *expl3*, it depends on the \LaTeX 3 group of packages *l3kernel* and *l3package*. In addition, *intexgral* uses the *derivative*¹ package to facilitate the manipulation of differentials, as well as *pkginfograb*².

*E-mail: vdao.texdev@gmail.com

¹Available on CTAN under: <https://ctan.org/pkg/derivative>

²Available on CTAN under: <https://ctan.org/pkg/pkginfograb>

Contents

Abstract	1
1 Package options	4
2 Presentation of the main macro	4
3 List of keys	5
3.1 Integration limits	5
3.1.1 Defining the limits	5
3.2 Display modes	6
3.3 Symbol (and more about limits)	7
3.3.1 Selecting a symbol	7
3.3.2 Generating many integrals	7
3.3.3 Defining boundaries on an ad hoc basis	8
3.3.4 Recurring limits motif	8
3.4 Differentials	9
3.4.1 Specifying the integration variables	9
3.4.2 Including the Jacobian	9
3.4.3 Modifying the differential symbol	9
3.4.4 Creating a path integral	10
3.4.5 Opting for the starred variant	10
3.4.6 Designating options	10
4 Additional macros	10
4.1 Custom differentials placement	11
4.2 Looking back at the <i>limits</i> key	12
4.3 Looking back at the <i>variables</i> key	13
4.4 Looking back at the <i>symbol</i> key	14
5 <i>Special</i> syntax	15
5.1 Presentation of the syntax	15
5.2 Functioning of the syntax	16
6 Additional parameters	16
Changelog	18
Index	20
Implementation	26

License

© 2025 Valentin Dao, published under the ~~La~~TeX Project Public License (LPPL) 1.3c

Fonts

Chronicle Text G3 Roman

© 2002, 2007 Hoefler & Frere-Jones.

Whitney-Medium

© 1996, 2009 Hoefler & Frere-Jones.

Monospace Argon Medium

© 2023, GitHub <https://github.com/githubnext/monospace>

Repository

 See GitHub repository.

Acknowledgements

Many thanks to Plante and Slurpy for accompanying me on this adventure of learning T_EX, your pieces of advice were priceless. I'd also like to thank Anthony, who has always kindly reviewed the new versions and given me ideas for those to come.

1 Package options

`\intexgralsetup` `\intexgralsetup` {<package options>}

new: v2.0.0

These options can be declared in a classic way with `\usepackage` or with `\intexgralsetup` in the preamble.

`invert-limits` `invert-limits=<true|false>`

This key inverts the limit order convention. The entirety of the document can only follow one convention.

`invert-diff` `invert-diff=<true|false>`

updated: v3.0.0

It is common to see the differentials placed before the integrand in physics papers. This key hence swaps their position.

`limits-mode` `limits-mode=<limits|nolimits>`

new: v3.0.0

Applies `\limits` or `\nolimits` to every single integral.

`italic` `italic=<true|false>`

`upright` `upright=<true|false>`

These are the two keys of the *derivative* package.

2 Presentation of the main macro

`\integral` `\integral` [<list of keys>] {<integrand>}

updated: v3.0.0

This macro is used to typeset integrals. It must, of course, be used in math mode only. Here is a first example in its simplest form:

```
\begin{equation}
  \integral{x}
\end{equation}
```

$$\int x \, dx \tag{1}$$

Since this macro focuses on the use of keys, the first part of this documentation will present them, grouping them by domain. The second part will introduce a few additional macros that complement the use of certain keys. The rest of this document will outline the other features of the package.

3 List of keys

3.1 Integration limits

3.1.1 Defining the limits

`limits` `limits=` {<mixed-list>}, <keyword>
`limits*` `limits*=` {<mixed-list>}, <keyword>

This key determines the integration limits to be used, following the convention explained in the abstract. In its simplest form, the value of the key takes as its argument a list of elements separated by commas (*comma-separated values* or <csv-list>¹).

```
\begin{equation}
  \integral[limits={1, 10}]{f(x)}
\end{equation}
```

$$\int_1^{10} f(x) \, dx \tag{2}$$

The notable advantage of the `limits` key is that you can specify the integration limits for several integrals, and the package automatically typesets the corresponding symbols. To separate pairs of limits, we use a semicolon (*semicolon-separated values* or <ssv-list>¹).

```
\begin{equation}
  \integral[limits={1, 2; 3, 4}, variables={x, y}]{f(x, y)}
\end{equation}
```

$$\int_1^2 \int_3^4 f(x, y) \, dx \, dy \tag{3}$$

It is also possible to specify predefined limits using keywords (see section 4.2). Finally, the starred variant allows the limits of the integral to be expressed as an interval. The direction of the brackets automatically adjusts itself to the presence of $\pm\infty$.

```
\begin{equation}
  \integral[limits*={1, 10}]{f(x)}
\end{equation}
```

$$\int_{[1,10]} f(x) \, dx \tag{4}$$

¹We therefore mean by <mixed-list> that `limits` can accept a set of <csv-list> in a more general <ssv-list>.

3.2 Display modes

mode `mode=<default|nested|product>`

new: v3.0.0 *Solely* when the `limits` key is used, it is possible to choose between three distinct display styles. Each of these is fully compatible with the `invert-diff` option.

default Typesets all the symbols, then the integrand, then the variables. As the macro operates by default in this mode, it is not necessary to use the key with this value.

nested Typesets a nested integral where the symbol and integrand alternate before all the variables are written.

product Typesets a product of integrals where the symbol, integrand, and variables alternate.

Of course, nothing prevents you from using the `\integral` macro several times in a row to produce the same result as in the example below. However, for those who would prefer to obtain the result with a single macro, this method is permitted.

Important: for the last two modes, the integral will be *cut off* in a certain way. In order to perform this action correctly, the same method must be used as with `limits`, i.e. by using a semicolon.

MODE **default**

```
\begin{equation}
  \integral[limits={1, 2; 3, 4; 5, 6}, variables={x, y, z}, mode=default]{xyz}
\end{equation}
```

$$\int_1^2 \int_3^4 \int_5^6 xyz \, dx \, dy \, dz \quad (5)$$

MODE **nested**

```
\begin{equation}
  \integral[limits={1, 2; 3, 4; 5, 6}, variables={z, y, x}, mode=nested]{x;y;z}
\end{equation}
```

$$\int_1^2 x \int_3^4 y \int_5^6 z \, dz \, dy \, dx \quad (6)$$

MODE **product**

```
\begin{equation}
  \integral[limits={1, 2; 3, 4; 5, 6}, variables={x, y, z}, mode=product]{x;y;z}
\end{equation}
```

$$\int_1^2 x \, dx \int_3^4 y \, dy \int_5^6 z \, dz \quad (7)$$

To function correctly, it is obvious that the keys `limits` and `variables` (see 3.4.1), as well as the integrand, must have the same number of elements. If this is not the case,

compilation will not fail, but the expression of the integral may no longer make sense. Numerous warning messages will be displayed to inform you of this.

3.3 Symbol (and more about limits)

The `limits(*)` keys are very useful when typesetting a definite integral. However, this only covers final expressions where the limits of each variable have been specified. For more general cases — indefinite integrals — they are relatively inconvenient. To typeset a double integral over a surface S , one would have to write `limits={,;S,}`. Needless to say, this is rather inconvenient for the user and not optimal for the package. Furthermore, the glyph would not be correct. The set of keys below therefore offers an easy way to modify both the symbol and the bounds.

3.3.1 Selecting a symbol

`symbol` `symbol=<control sequence>`

updated: v3.0.0 This key accepts a macro designating an integral symbol. Any symbol previously defined by a control sequence is acceptable. If you attempt to use one that is not defined, it will be substituted with `\int` and a warning message will be issued.

Remark: apart from verifying the existence of the macro, no specific checks are performed, and the value of the key is used as is to compose the symbol. This implies two things in particular. First, the symbol will depend on how it is defined. For instance, *amsmath* and *unicode-math* do not define `\int` in the same way. The result will naturally be different. Furthermore, if you assign a *defined* macro as the key argument but it does not correspond to an integral symbol, the package will ignore it and still use the given symbol. In addition to the mathematical inconsistency that this may represent, it is likely to lead, under certain circumstances, to low-level errors.

```
\begin{equation}
  \integral[symbol=\iint, llimit=S, variables={x, y}]{f(x, y)}
\end{equation}
```

$$\iint_S f(x, y) \, dx \, dy \quad (8)$$

3.3.2 Generating many integrals

`nint` `nint=<integer>`

This key accepts an integer n which allows n integrals to be composed. It is advisable to use this key only if the number of symbols exceeds 4 to favour the glyphs defined by the mathematical font used.

```
\begin{equation}
  \integral[nint=5, llimit=\Omega, variables={x_1, x_2, x_3, x_4, x_5}]{f(x_1, x_2,
  x_3, x_4, x_5)}
\end{equation}
```

$$\iiint\limits_{\Omega} f(x_1, x_2, x_3, x_4, x_5) \, dx_1 \, dx_2 \, dx_3 \, dx_4 \, dx_5 \quad (9)$$

3.3.3 Defining boundaries on an ad hoc basis

`llimit` `llimit=<lower limit>`
`ulimit` `ulimit=<upper limit>`

updated: v3.0.0

These two keys allow you to specify the lower and upper limits, respectively. They are only suitable if a single symbol is displayed. If you need to specify both limits, the `limits` key should be used instead.

```
\begin{equation}
\integral[llimit={x^2 + y^2 \leq 1}, variables={x, y}]{f(x, y)}
\end{equation}
```

$$\int_{x^2+y^2 \leq 1} f(x, y) \, dx \, dy \quad (10)$$

3.3.4 Recurring limits motif

Lower limits sometimes follow common patterns that can be generalised with keys, thus avoiding overloading the argument of `llimit`.

`domain` `domain= {<*-list>}`
`domain*` `domain*= {<*-list>}`

updated: v3.0.0

These keys accept a list delimited by an asterisk. Each element of the list is then parsed as follows:

- The first token of the item is passed `\mathbb` (preceded by `\uppercase` in case the SHIFT key is too far away for your fingers).
- The remaining tokens — which may be empty — are placed as superscript (or subscript for the starred variant of the key).

```
\begin{equation}
\integral[symbol=\iint, domain={r*r}, variables={x, y}]{xy}
\end{equation}
```

$$\iint_{\mathbb{R} \times \mathbb{R}} xy \, dx \, dy \quad (11)$$

`boundary` `boundary= {<lower limit>}`

This key simply places the symbol ∂ before the lower bound.

```
\begin{equation}
\integral[symbol=\oint, boundary=S, diff-vec]{G(\vec{r})}
\end{equation}
```

$$\oint_{\partial S} G(\vec{r}) \cdot d\vec{r} \quad (12)$$

3.4 Differentials

3.4.1 Specifying the integration variables

variables `variables={⟨csv-list⟩}, ⟨keyword⟩, none`

updated: v3.0.0

This key allows you to define the variables of the integral in the form of a `⟨csv-list⟩`. Like `limits`, the key can accept a keyword as an argument. This behaviour is also explained later (see section 4.3).

Remark: if no variable is specified, i.e. `variables` is not called, the package automatically sets “`dx`” (or “`d \vec{r}` ” if `diff-vec` is active). Furthermore, if `none` is passed as a value, no variable will be displayed.

```
\begin{equation}
  \integral[variables=t]{t^2}
\end{equation}
```

$$\int t^2 dt \quad (13)$$

3.4.2 Including the Jacobian

jacobian `jacobian`

This key enables the display of the Jacobian when defined by `\NewVariableKeyword`.

```
\begin{equation}
  \integral[limits={0, R; 0, 2\pi; 0, \pi}, variables=spherical, mode=product,
  jacobian]{;}
\end{equation}
```

$$\int_0^R r^2 dr \int_0^{2\pi} \sin \theta d\theta \int_0^\pi d\phi \quad (14)$$

As mentioned earlier, `derivative` is used by the package to compose differentials. The set of keys to follow, whose names will be prefixed with “`diff-`”¹, thus allows you to apply the features of the macro `\odif`.

3.4.3 Modifying the differential symbol

diff-symb `diff-symb=⟨differential macro⟩`

This key allows you to change the style of differentials among those defined by `\NewDifferential`.

```
\NewDifferential{\feynmandiff}{\mathcal{D}}
\begin{equation}
  \integral[variables={q(t)}, diff-symb=\feynmandiff]{e^{\dfrac{\imath}{\hbar} S[q(t)]}}
\end{equation}
```

¹Although the key `diff-vec` is not linked to `derivative`.

$$\int e^{+\frac{iS[q(t)]}{\hbar}} \mathcal{D}q(t) \quad (15)$$

3.4.4 Creating a path integral

diff-vec diff-vec

This key applies a vector to each variable in addition to a centred dot. It is only compatible with the `default` mode.

```
\begin{equation}
\integral[double=S, variables=S, diff-vec]{\vec F}
\end{equation}
```

$$\iint_S \vec{F} \cdot d\vec{S} \quad (16)$$

3.4.5 Opting for the starred variant

diff-star diff-star

Activating this key is equivalent to using `\odif*` or its variants.

```
\begin{equation}
\integral[double, variables={x, y}, diff-star]{x^2 \exp(y)}
\end{equation}
```

$$\iint x^2 \exp(y) \, d_x d_y \quad (17)$$

3.4.6 Designating options

diff-options diff-options={keyval list}

This key accepts a list of keys as it would have been written as an optional argument of `\odif` or its variants. For example, `diff-options={order={2, 3}, var=all}` will be interpreted as `\odif[order={2, 3}, var=all]{...}`.

```
\begin{equation}
\integral[diff-options={order=4}]{\bar{\psi}(x)(\imath\gamma^\mu\partial_\mu-m)\psi(
x)}
\end{equation}
```

$$\int \bar{\psi}(x)(i\gamma^\mu\partial_\mu - m)\psi(x) \, d^4x \quad (18)$$

4 Additional macros

In addition to the wide range of keys defined by the package, a few macros also enhance their uses.

4.1 Custom differentials placement

`\differentials` `\differentials`

Although the `invert-diff` option exists, it is often desirable to be able to place differentials wherever one wishes. For example, they are frequently seen in the numerator of a fraction when the numerator is 1. To meet this need, the package provides the macro `\differentials`. For the `default` and `nested` modes, the macro typesets all differentials at once. With `product`, the macro must be repeated between each semicolon.

```
\begin{equation}
  \integral{\frac{\differentials}{x}}{}
\end{equation}
```

$$\int \frac{dx}{x} \tag{19}$$

4.2 Looking back at the *limits* key

```
\NewLimitsKeyword    \NewLimitsKeyword {\keyword} {\limits}
\RenewLimitsKeyword
\ProvideLimitsKeyword
\DeclareLimitsKeyword
```

It is common to have to specify the same limits in an integral. To make them easier to write, the `limits(*)` keys accept, in addition to explicit limits, a keyword designating a pair of them predefined by one of these four macros:

- `\NewLimitsKeyword` Creates a new keyword and issues an error if it already exists.
- `\RenewLimitsKeyword` Redefines a keyword and issues an error if it does not already exist.
- `\ProvideLimitsKeyword` Only creates a new keyword if it does not already exist. No message will be issued if this is the case.
- `\DeclareLimitsKeyword` Creates a new keyword regardless, overwriting any previous definition.

Here is the list of keywords already defined by the package and the limits they contain:

<code>ab</code>	a, b
<code>unit</code>	$0, 1$
<code>real</code>	$-\infty, +\infty$
<code>positive</code>	$0, +\infty$
<code>negative</code>	$-\infty, 0$
<code>circle</code>	$0, 2\pi$
<code>sircle</code>	$0, \pi$
<code>qcircle</code>	$0, \frac{\pi}{2}$
<code>height</code>	$0, H$
<code>radius</code>	$0, R$
<code>length</code>	$0, L$
<code>time</code>	$0, T$

Remark: keywords contain both limits of an integral at the same time. They must therefore be preceded and/or followed by semicolons.

Important: due to the implementation of `\NewLimitsKeyword` and its variants, the user *must* follow the package's limit order convention, even if `invert-limits` is set to `true`

We could therefore modify the expression of an integral as follows:

```
\begin{equation}
  \integral[limits={radius;circle;scircle}, variables={\rho, z, \phi}]{\rho z \phi}
\end{equation}
```

$$\int_0^R \int_0^{2\pi} \int_0^\pi \rho z \phi \, d\rho \, dz \, d\phi \quad (20)$$

It is entirely possible to combine these keywords with the more explicit syntax of `limits`.

```
\begin{equation}
  \integral[limits={time; 34, 40; height}, variables={h, \omega, t}]{h \omega t}
\end{equation}
```

$$\int_0^T \int_{34}^{40} \int_0^H h \omega t \, dh \, d\omega \, dt \quad (21)$$

4.3 Looking back at the *variables* key

```

\NewVariableKeyword    \NewVariableKeyword {\keyword} {\variables} [\jacobian]
\RenewVariableKeyword
\ProvideVariableKeyword
\DeclareVariableKeyword

```

Similarly, the same groups of variables often reappear. We can therefore also define keywords containing a set of pre-registered variables. The variants of this macro behave in the same way as for `\NewLimitsKeyword`. The only difference is that here it is possible to define a Jacobian, in the form of a `\csv-list` if necessary. Its display is then controlled by the `jacobian` key as explained above. Here is the list of variable keywords already defined by the package, with the Jacobian for some of them:

```

cartesian    x, y, z
planar       x, y
polar        r, \theta(r)
cylindrical  r, \theta, z(r)
spherical    r, \theta, \phi(r^2, \sin \theta)

```

```

\begin{equation}
  \integral[triple=V, variables=spherical]{f(x, y, z)}
\end{equation}

```

$$\iiint_V f(x, y, z) \, dr \, d\theta \, d\phi \quad (22)$$

4.4 Looking back at the *symbol* key

```
\NewSymbolKeyword    \NewSymbolKeyword {⟨key⟩} {⟨symbol⟩}
\RenewSymbolKeyword
\ProvideSymbolKeyword
\DeclareSymbolKeyword
```

new: v3.0.0

In order to typeset indefinite integrals, the package essentially offers two keys: `symbol` and `llimit`. Although they are simple to use, it is still laborious to write both of them with their values. This is why *intexgral* provides so-called *shortcut* keys, whose purpose is to combine the selection of the symbol and the limits. Unlike the two previous macros, this involves creating entirely new keys, rather than simply specific values assigned to `symbol`. On their own, these keys modify the integral symbol. The value of these keys corresponds to the lower bound. Here is the list of all *symbol-key* defined by the package:

<code>single</code>	used symbol = <code>\int</code>
<code>double</code>	used symbol = <code>\iint</code>
<code>triple</code>	used symbol = <code>\iiint</code>
<code>quadruple</code>	used symbol = <code>\iiiiint</code>
<code>contour</code>	used symbol = <code>\oint</code>
<code>surface</code>	used symbol = <code>\oiint</code>
<code>volume</code>	used symbol = <code>\oiiint</code>

Important: allowing users to create their own keys for the `\integral` macro poses a risk that will be explained in more detail in the next section. For now, just remember that all these macros¹ will issue a warning message if you attempt to create a new *symbol-key* with the same name as a group of defined limits.

```
\begin{equation}
  \integral[contour=\mathcal{C}, diff-vec]{f(\vec{r})}
\end{equation}
```

$$\oint_C f(\vec{r}) \cdot d\vec{r} \quad (23)$$

5 Special syntax

5.1 Presentation of the syntax

`\integral` `\integral` [`\limits:variables(+j):mode`] {`\integrand`}

new: v3.0.0

With regard to definite integrals, the user will be required to use a maximum of four keys to compose them: `limits`, `variables`, `mode` and `jacobian`. However, they can be criticised for the same reason as before; writing these four keys, which can take quite long arguments, runs counter to the main objective of this package. Thus, the user can designate as an optional argument of `\integral` a syntax called *special* that is unique to *intexgral*. The logic is as follows:

- You specify a *valid* argument for the `limits` key.
- You specify a *valid* argument for the `variables`
- You specify a *valid* argument for the `mode` key

And you separate everything with a colon. Let's look at an example to better understand.

```
\begin{equation}
  \integral[1, 2; 4, 5:y, x:nested]{x; y}
\end{equation}
```

$$\int_1^2 x \int_4^5 y \, dy \, dx \quad (24)$$

We mean by *valid argument* that all forms of values accepted by the four keys can also be likewise adopted by the special syntax. This hence includes keywords.

```
\begin{equation}
  \integral[radius;circle;scircle:spherical:product]{r;\theta;\phi}
\end{equation}
```

$$\int_0^R r \, dr \int_0^{2\pi} \theta \, d\theta \int_0^\pi \phi \, d\phi \quad (25)$$

To include the Jacobian, simply write `+j` after the argument of `variables`. The mode can also be indicated using only an initial.

```
\begin{equation}
  \integral[radius;circle;scircle:spherical+j:p]{;};
\end{equation}
```

$$\int_0^R r^2 \, dr \int_0^{2\pi} \sin \theta \, d\theta \int_0^\pi d\phi \quad (26)$$

It is important to reiterate that in the classic configuration of the optional argument, it is of course not mandatory to specify the four keys mentioned. The same applies to this syntax. Since the key `mode` is not necessary for `default`, this part can be omitted.

```
\begin{equation}
  \integral[1, 2; 3, 4:x, y]{xy}
\end{equation}
```

$$\int_1^2 \int_3^4 xy \, dx \, dy \quad (27)$$

You can also take advantage of the automatic placement of “dx” with this syntax, ignoring `variables`.

```
\begin{equation}
  \integral[1, 10]{x}
\end{equation}
```

$$\int_1^{10} x \, dx \quad (28)$$

5.2 Functioning of the syntax

From the user’s point of view, it is relatively simple to choose which syntax to adopt. However, the package must be able to distinguish between the two. Without going into too much detail¹ about the implementation, it is brought to your attention that the package attempts to extract the first key from the optional argument² and checks whether it exists. This is why, if you create a key with `\NewSymbolKeyword` whose name can probably be used as integration limits, *intexrgal* may incorrectly evaluate the nature of the optional argument.

6 Additional parameters

`\IntegralSetup` `\IntegralSetup {⟨parameter list⟩}`

new: v3.0.0

This macro allows you to modify parameters related to certain keys using the syntax `⟨key=value⟩`. All assignments made are local, so the macro can be placed in a group if necessary. Here is an example of the macro with the package’s default assignments:

```
\IntegralSetup
{
  defaultvar = {x},
  defaultvar* = {r},
  vectorstyle = \vec,
  domainstyle = \mathbb,
  symbolskip = {-6mu},
  hide-diff = false
}
```

`defaultvar` `defaultvar= {⟨variables⟩}`

`defaultvar*`

This key modifies the variable automatically inserted when `variables` is not used. The argument may correspond to a keyword defined by `\NewVariableKeyword`. The starred variant of the key controls the integration parameter to be typeset with `diff-vec`.

¹The more courageous among you are still invited to read the source code.

²Or at least, the group of tokens that it thinks corresponds to a key.

vectorstyle `vectorstyle=<control sequence>`

This key modifies the macro to be applied to variables when `diff-vec` is in action. It is therefore possible to alter the style of the vector: bold, underlined, or even another style from a particular package, `esvect` for example.

domainstyle `domainstyle=<control sequence>`

Similarly, you can change the macro to be applied for keys `domain(*)`.

symbolskip `symbolskip=<muexpr>`

When the `default` mode is in effect, the package inserts kerning between symbols to bring them slightly closer together. The default dimension (-6μ) can thus be adjusted according to the mathematical font in use.

hide-diff `hide-diff=<true|false>`

When a large part of the document requires differentials not to be included, it is possible to extend the more specific action of `variables=none` over time.

Changelog

3.0.1 (02-01-2026)

Fixed

- ▶ Bug with jacobian in special syntax ([issue #3](#)).
- ▶ French and English documentations ([issue #4](#), [issue #6](#) and [issue #7](#)).

Changed

- ▶ Limits keywords `positive` and `real` now contain a + sign ([issue #5](#)).

3.0.0 (24-12-2025)

Added

- ▶ Special syntax.
- ▶ Keys `domain*` and `mode`.
- ▶ Macros `\IntegralSetup` and `\NewSymbolKeyword`.

Removed

- ▶ Macros `\defaultdiff`, `\defaultdiff`, `\defaultvdiff` et `\vdiffstyle` in favour of `\IntegralSetup`.
- ▶ Keys controlling both symbol and limit, now managed at a higher level by `\NewSymbolKeyword`.
- ▶ `int-split` key, in favour of `mode`.
- ▶ `\NewIntegralSymbol` macro.

Changed

- ▶ The names of some keys (`variable` in `variables`, `lower-lim` and `upper-lim` in `llimit` and `ulimit`, `int-symb` in `symbol`, `invert-differentials` and `hide-differentials` in `invert-diff` and `hide-diff`).
- ▶ `jacobian` and `diff-star` no longer require a boolean value. Simply writing them will now enable their features.
- ▶ `hide-diff` assigned to a local option rather than a package one.
- ▶ `limits-mode` assigned to a package option rather than a macro key

v2.0.1 (13-09-2025)

Fixed

- ▶ Compatibiliy issue between unicode-math and amssyb depending on the loading order ([problem #2](#)).

2.0.0 (09-09-2025)

Added

- ▶ Macros `\intexgralsetup`, `\defaultdiff`, `\defaultvdiff` and `\vdiffstyle`.

Changed

- ▶ Warning messages related to non-existing symbols. They are now only triggered when the integral is typeset.

Removed

- ▶ `diff-vec-style` key in favour of `\vdiffstyle`.
- ▶ Warning message about misuse of semi-colon in conjunction with the `int-split` key.

Fixed

- ▶ Bug where the integrand was not reset when `\integral` was used successively in the same \TeX group ([details here](#)).
- ▶ `expl3` log leftovers that shouldn't have been there.

1.1.0 (29-07-2025)

Added

- ▶ Starred variant for the keys controlling both symbol and limit (keys `single`, `contour` etc).

1.0.0 (26-07-2025) — Initial version.

Index

Symbols

\! 751, 791, 839
 \: 752
 \{ 419, 434, 459, 461
 \} 419, 434, 459, 461
 _ . 97, 101, 105, 109, 113, 117, 122, 150, 152

B

bool commands:

\bool_if:NTF 294, 352, 369, 409, 485, 493, 549, 556, 572, 701, 722, 729, 737, 752, 755, 764, 773, 778, 783, 785, 795, 799, 802, 806, 817, 825, 842, 1029, 1057, 1071, 1086, 1100
 \bool_if:nTF 654, 744, 757, 830, 844
 \bool_new:N 39, 40, 41, 201, 202, 203, 204, 205, 206, 207, 208
 \bool_set_false:N 210, 211, 212, 213, 214, 215, 727, 878
 \bool_set_true:N 421, 437, 442, 726, 883, 888, 999, 1001, 1020, 1022, 1024

C

\c 725
 \cdot 752, 764
 clist commands:
 \clist_item:Nn .. 338, 340, 356, 358, 364, 366, 373, 379, 597
 \clist_map_inline:Nn 551
 \clist_new:N 217, 218
 \clist_put_right:Nn 553
 \clist_reverse:N .. 1060, 1074, 1089, 1103
 \clist_set:Nn 335, 350, 398, 422, 445, 471, 596, 1006, 1015, 1059, 1073, 1088, 1102
 \l_tmpa_clist 335, 338, 340, 350, 356, 358, 364, 366, 373, 379, 398, 401, 422, 426, 445, 449, 471, 472, 596, 597, 1059, 1060, 1061, 1073, 1074, 1075, 1088, 1089, 1090, 1102, 1103, 1104

cs commands:

\cs_generate_variant:Nn . 172, 173, 174, 175, 176, 177, 178
 \cs_if_exist:NTF 35, 913

\cs_new:Nn 86
 \cs_new:Npn 312, 407, 590
 \cs_new_protected:Nn . 267, 331, 669, 697, 735, 771, 815, 856
 \cs_new_protected:Npn 245, 248, 255, 292, 307, 321, 346, 392, 416, 431, 454, 469, 547, 594, 611
 \cs_set:Npn 1028
 \cs_set_eq:NN 788
 \cs_set_protected:Npn 739, 775, 819

D

\d 419, 434, 459, 461
 \DeclareLimitsKeyword 1096
 \DeclareSymbolKeyword 1204
 \DeclareVariableKeyword 1138
 \differentials 150, 152, 739, 775, 788, 819

E

exp commands:

\exp_args:Ne 959, 980
 \exp_args:NNne 621
 \exp_args:NV ... 275, 563, 579
 \exp_args:NVV 558, 574
 \exp_last_unbraced:NV ... 602
 \exp_not:N 964, 985
 \exp_not:n 538
 \ExplSyntaxOff 1281

F

\frac 1256

G

group commands:

\group_begin: 1222
 \group_end: 1230

H

hook commands:

\hook_gput_code:nnn 34

I

\iiiint 1243
 \iiint 1242
 \iint 1241
 \infty ... 375, 381, 1250, 1251, 1252
 \int 131, 199, 917, 933, 1240
 int commands:
 \int_compare:nNnTF . 495, 625, 632, 671, 676, 925, 934

<code>\int_compare_p:nNn</code>	656	<code>__integral_differentials:nn</code>	
<code>\int_gincr:N</code>	699	407, 537, 1028
<code>\int_gzero_new:N</code>	241	<code>\l__integral_display_mode_-</code>	
<code>\int_new:N</code>	240	<code>str</code>	233, 859, 879, 884, 889
<code>\int_set:Nn</code>	269	<code>\l__integral_domain_char_tl</code>	
<code>\int_step_inline:nn</code> .	276, 780, 827, 930	. .	192, 958, 965, 979, 986
<code>\int_use:N</code>	89	<code>\l__integral_domain_dimension_-</code>	
<code>\integral</code>	1220	<code>tl</code>	193, 960, 966, 981, 987
<code>\IntegralSetup</code>	1217, 1271	<code>\l__integral_domain_seq</code>	231, 953, 954, 974, 975
integral internal commands:		<code>\l__integral_domain_style_tl</code>	184, 964, 985, 1043
<code>__integral_check_sequence_-</code>		<code>__integral_extract_differential_-</code>	
<code>size:</code>	669, 730	<code>order:n</code>	416, 475
<code>\l__integral_custom_variables_-</code>		<code>__integral_extract_differential_-</code>	
<code>bool</code>	202, 212, 556, 572, 999	<code>var:n</code>	431, 476
<code>\l__integral_deactivate_-</code>		<code>__integral_extract_first_-</code>	
<code>differentials_bool</code>	40, 722, 1001, 1045	<code>key_name:n</code>	594, 1225
<code>\l__integral_default_differential_-</code>		<code>__integral_generate_integral_-</code>	
<code>tl</code>	194, 581, 622, 629, 1040	<code>sequence:</code>	267, 897, 904
<code>\l__integral_default_vector_-</code>		<code>\l__integral_has_order_bool</code>	
<code>differential_tl</code> . .	195, 567, 1041	207, 214, 421, 485
<code>\g__integral_differential_-</code>		<code>\l__integral_has_var_bool</code> . .	208, 215, 437, 442, 493
<code>group_keyword_prop</code> . .	238, 394, 1004, 1114, 1122, 1133, 1140	<code>\l__integral_independent_-</code>	
<code>\l__integral_differential_-</code>		<code>differential_options_tl</code> .	
<code>options_i_tl</code>	188, 559, 564, 575, 580, 1035	191, 456, 464, 484
<code>\l__integral_differential_-</code>		<code>\l__integral_inline_symbol_-</code>	
<code>options_ii_tl</code>	189, 483, 487, 501, 515, 527, 538, 541	<code>muskip</code>	235, 743, 1044
<code>\l__integral_differential_-</code>		<code>\g__integral_integral_-</code>	
<code>order_i_seq</code>	229, 420, 423	<code>number_int</code>	89, 241, 699
<code>\l__integral_differential_-</code>		<code>__integral_integral_preconfiguration:</code>	
<code>order_ii_seq</code> . . .	230, 425, 490	697, 858
<code>\l__integral_differential_-</code>		<code>\l__integral_integral_-</code>	
<code>seq</code>	223, 535, 679, 684, 691, 740, 751, 765, 776, 790, 808, 838, 850, 866	<code>sequence_int</code> . . .	240, 269, 276
<code>\l__integral_differential_-</code>		<code>\l__integral_integral_-</code>	
<code>var_i_seq</code>	227, 436, 441, 446, 496, 506, 510	<code>symbol_seq</code>	221, 278, 672, 682, 689, 709, 711, 742, 780, 782, 827, 829
<code>\l__integral_differential_-</code>		<code>\l__integral_integral_-</code>	
<code>var_ii_seq</code> . .	228, 448, 520, 524	<code>symbol_tl</code>	187, 199, 257, 914, 917, 928, 932, 936

<code>\l__intexgral_integrand_seq .</code>	<code>limit:n</code>	312, 336, 351
<code>. 220, 674, 677, 683, 690, 704, 715, 718, 754, 794, 801, 841</code>	<code>__intexgral_parse_integral_-</code>	
<code>\l__intexgral_integrand_tl 183,</code>	<code>symbol:</code>	255, 283, 712
<code>706, 719, 725, 1228</code>	<code>__intexgral_parse_limits:n .</code>	
<code>\l__intexgral_invert_differentials_-</code>	<code>.</code>	321, 895, 902
<code>bool 41, 52, 746, 759, 783, 799,</code>	<code>__intexgral_parse_macro_-</code>	
<code>832, 846</code>	<code>keys:n</code>	611, 1226
<code>\l__intexgral_invert_limits_-</code>	<code>__intexgral_parse_variable:nn</code>	
<code>bool . . 39, 48, 294, 352, 369,</code>	<code>. . 469, 558, 563, 574, 579</code>	
<code>1057, 1071, 1086, 1100</code>	<code>__intexgral_print_default_-</code>	
<code>\l__intexgral_jacobian_bool .</code>	<code>integral:</code>	735, 861, 865
<code>205, 755, 795, 802, 842, 1020</code>	<code>__intexgral_print_integral:</code>	
<code>\l__intexgral_jacobian_seq . .</code>	<code>.</code>	856, 1229
<code>222, 756, 796, 803, 843, 1010</code>	<code>__intexgral_print_nested_-</code>	
<code>\l__intexgral_key_name_tl 182,</code>	<code>integral:</code>	771, 862
<code>600, 606, 613</code>	<code>__intexgral_print_product_-</code>	
<code>\c__intexgral_left_bracket_tl</code>	<code>integral:</code>	815, 863
<code>. 196, 377, 381</code>	<code>__intexgral_remove_differential_-</code>	
<code>\g__intexgral_limits_keyword_-</code>	<code>order_and_var:n</code>	454, 477
<code>prop 237,</code>	<code>__intexgral_retrieve_key_-</code>	
<code>308, 314, 315, 1054, 1069, 1084,</code>	<code>name:w</code>	590, 603
<code>1098, 1152, 1170, 1189</code>	<code>\c__intexgral_right_bracket_-</code>	
<code>\l__intexgral_limits_seq 224,</code>	<code>tl 197, 375,</code>	383
<code>326, 333, 348</code>	<code>\l__intexgral_separate_-</code>	
<code>\l__intexgral_limits_style_tl</code>	<code>integral_bool 203, 213, 701,</code>	729, 878, 883, 888
<code>. 42, 58, 60, 258</code>	<code>__intexgral_set_limits:nn . .</code>	
<code>\l__intexgral_lower_limit_tl</code>	<code>. 280, 292</code>	
<code>. 185, 260, 296,</code>	<code>__intexgral_set_limits_-</code>	
<code>300, 354, 362, 378, 907, 956,</code>	<code>regular:</code>	331, 896
<code>957, 962, 977, 978, 983, 993</code>	<code>__intexgral_set_limits_-</code>	
<code>\l__intexgral_lower_limits_-</code>	<code>starred:</code>	346, 903
<code>seq . . 226, 271, 273, 281, 324,</code>	<code>\l__intexgral_starred_differential_-</code>	
<code>337, 371</code>	<code>bool 206, 409, 1024,</code>	1029
<code>\l__intexgral_manual_differentials_-</code>	<code>\l__intexgral_upper_limit_tl</code>	
<code>bool 201,</code>	<code>. . 186, 262, 297, 301, 909</code>	
<code>211, 726, 727, 737, 748, 761, 773,</code>	<code>\l__intexgral_upper_limits_-</code>	
<code>785, 806, 817, 834, 848</code>	<code>seq . 225, 272, 282, 325, 339,</code>	370
<code>__intexgral_mathchoice:nnnn</code>		
<code>. 248, 938</code>		
<code>__intexgral_mkern:N . 245, 743</code>		
<code>__intexgral_new_differential_-</code>		
<code>group:nnn . . 392, 1117, 1126,</code>		
<code>1135, 1144</code>		
<code>__intexgral_new_limits_-</code>		
<code>group:nn 173, 307, 1061,</code>		
<code>1063, 1075, 1077, 1090, 1092,</code>		
<code>1104, 1106</code>		
<code>__intexgral_parse_differentials:</code>		
<code>. 547, 723</code>		
<code>__intexgral_parse_integral_-</code>		

`\l__intexgral_variable_i-`
`clist .. 217, 551, 576, 1006,`
`1015`
`\l__intexgral_variable_ii-`
`clist 218, 553,`
`560`
`\l__intexgral_vectorial_-`
`differential_bool 204, 210,`
`549, 752, 764, 778, 825, 1022`
`\l__intexgral_vectorial_-`
`differential_style_tl 190,`
`554, 566, 1042`
`__intexgral_warning_msg_-`
`header: 86, 122, 129, 136, 143,`
`151, 159, 165`
`\intexgralsetup 1235`
iow commands:
`\iow_newline: 22, 23, 92`

K

keys commands:
`\keys_define:nn .. 46, 871, 1038,`
`1157, 1174, 1193, 1206`
`\keys_if_exist:nnTF . 176, 613,`
`1154, 1172, 1191`
`\keys_set:nn ... 614, 637, 1218`

M

`\mathbb 32, 35, 1276`
`\mathchoice 249`
msg commands:
`\msg_critical:nn 30`
`\msg_error:nnn 1055, 1079, 1116,`
`1128, 1155, 1184`
`\msg_line_context: 90`
`\msg_new:nnn 19, 119, 127, 134, 141,`
`148, 157, 163`
`\msg_new:nnnn . 95, 99, 103, 107,`
`111, 115`
`\msg_see_documentation_text:n`
`..... 24`
`\msg_warning:nn 787, 926`
`\msg_warning:nnn 779, 826, 916,`
`1013, 1153, 1171, 1190`
`\msg_warning:nnnn .. 681, 688`
muskip commands:
`\muskip_new:N 235`

N

`\NeedsTeXFormat 3`
`\NewDocumentCommand 1052, 1067, 1082, 1096,`
`1112, 1120, 1131, 1138, 1150, 1168,`
`1187, 1204, 1217, 1220`
`\NewLimitsKeyword 117, 122, 1052, 1249, 1250, 1251,`
`1252, 1253, 1254, 1255, 1256,`
`1257, 1258, 1259, 1260`
`\NewSymbolKeyword 101, 1150, 1240, 1241, 1242,`
`1243, 1244, 1245, 1246`
`\NewVariableKeyword 1112, 1263, 1264,`
`1265, 1266, 1267`
`\NewVarKeyword 109`

O

`\odif 405, 410, 411`
`\oiint 1246`
`\oint 1245`
`\oint 1244`

P

`\partial 993`
`\PassOptionsToPackage 67, 69, 73, 75`
`\phi 1267`
`\pi 1254, 1255, 1256`
`\pkginfograbProvidesExplPackage 9`
`\ProcessKeyOptions 80`
prop commands:
`\prop_get:NnNTF ... 1003, 1054,`
`1084, 1114, 1133`
`\prop_if_in:NnTF 314, 1152, 1170,`
`1189`
`\prop_item:Nn 315`
`\prop_new:N 237, 238`
`\prop_pop:NnNTF 1069, 1098, 1122,`
`1140`
`\prop_put:Nnn 308, 394`
`\ProvideLimitsKeyword 1082`
`\ProvideSymbolKeyword 1187`
`\ProvideVariableKeyword 1131`

Q

quark commands:
`\q_stop 590, 603`

R

regex commands:
`\regex_extract_once:nnNTF 418,`
`433`
`\regex_if_match:nnTF . 172, 725`
`\regex_replace_all:nnN .. 457`
`\regex_split:nnN 616`
`\RenewLimitsKeyword 113, 1067`
`\RenewSymbolKeyword 97, 1168`

`\RenewVariableKeyword` 1120
`\RenewVarKeyword` 105
`\RequirePackage` 4, 5, 35, 82

S

`\s` 419, 434, 459, 461
 scan commands:
 `\scan_stop:` . 246, 788, 939, 940,
 941, 942
 seq commands:
 `\seq_clear:N` 324, 325, 1125, 1143
 `\seq_count:N` . . . 272, 273, 496,
 625, 632, 656, 672, 674, 677,
 679, 682, 683, 684, 689, 690,
 691, 780, 827
 `\seq_gclear:N` 866
 `\seq_gpop_left:NN` 821
 `\seq_if_empty:NTF` 271, 709, 715
 `\seq_if_exist:NTF` 1007
 `\seq_if_in:NnTF` 519
 `\seq_item:Nn` 281,
 282, 423, 446, 490, 506, 509,
 539, 618, 636, 639, 640, 643,
 658, 782, 790, 794, 796, 829,
 837, 841, 843, 850
 `\seq_map_indexed_inline:Nn` . .
 348, 479
 `\seq_map_inline:Nn` . . 333, 954,
 975
 `\seq_new:N` 220, 221, 222, 223, 224,
 225, 226, 227, 228, 229, 230,
 231
 `\seq_pop_left:NN` 523
 `\seq_put_left:Nn` 441
 `\seq_put_right:Nn` 278, 337, 339,
 370, 371, 535, 627, 633, 711, 717
 `\seq_set_eq:NN` 1009
 `\seq_set_from_clist:NN` . . 399,
 424, 447, 472
 `\seq_set_item:Nnn` 621
 `\seq_set_split:Nnn` . 326, 634,
 703, 953, 974
 `\seq_use:Nn` 177, 740,
 742, 751, 754, 756, 765, 776, 801,
 803, 808
 `\l_tmpa_seq`
 . . 472, 479, 539, 616, 618, 621,
 625, 628, 632, 633, 636, 639,
 643
 `\l_tmpb_seq` . 634, 640, 656, 658
`\sin` 1267

str commands:

`\str_case:nnTF` 643, 859
`\str_case_e:nnTF` 373, 379
`\str_if_eq:nnTF` . . 174, 175, 505,
 508, 617, 1000
`\str_if_eq_p:nn` 178, 658
`\str_if_in:nnTF` 439
`\str_new:N` 233
`\str_set:Nn` 879, 884, 889
`\str_uppercase:n` 959, 980

T

TeX and L^AT_EX 2_ε commands:

`\@ifpackagelater` 29
`\@onlypreamble` 1235

tex commands:

`\tex_left:D` 375, 377
`\tex_limits:D` 58
`\tex_mathop:D` . . . 751, 791, 839
`\tex_mathpunct:D` 357, 365
`\tex_mkern:D` 246, 939, 940, 941,
 942
`\tex_nolimits:D` 60
`\tex_right:D` 381, 383
`\theta` 1265, 1266, 1267
`\times` 957, 978

tl commands:

`\c_space_tl` 89
`\tl_clear:N` 541, 928
`\tl_const:Nn` 196, 197
`\tl_head:n` 959, 980
`\tl_if_blank:nTF` 396
`\tl_if_empty:NTF` 956, 977
`\tl_if_empty:nTF` 473, 480, 1223
`\tl_if_in:NnTF` 598
`\tl_new:N` . 42, 182, 183, 184, 185,
 186, 187, 188, 189, 190, 191, 192,
 193, 194, 195
`\tl_put_right:Nn` 482, 487,
 500, 514, 526, 932, 936, 957,
 962, 978, 983
`\tl_set:Nn` 296, 297,
 300, 301, 323, 354, 362, 456,
 597, 600, 952, 958, 960, 973,
 979, 981, 993, 1228
`\tl_set_eq:NN` . 58, 60, 199, 606,
 914, 917
`\tl_tail:n` 961, 982
`\tl_trim_spaces:n` 96, 100, 104,
 108, 112, 116, 130
`\tl_use:N` . . . 257, 258, 378, 822

<code>\l_tmpa_tl</code>	323, 326,		987
	525, 597, 598, 603, 606, 821,	<code>\c_math_superscript_token</code>	261,
	822, 952, 953, 973, 974, 1004,		966
	1006, 1054, 1069, 1084, 1098,	<code>\token_to_str:N</code>	97, 101, 105, 109,
	1115, 1123, 1134, 1141		113, 122, 131, 150, 152
token commands:		V	
<code>\c_math_subscript_token</code> .	259,	<code>\vec</code>	1275

Implementation

The implementation will be explained in more detail in a later version...

```
1 <*package>
2 <@@=intexgral>
3 \NeedsTeXFormat{LaTeX2e}
4 \RequirePackage{expl3}[2025-05-14]
5 \RequirePackage{pkginfograb}
6
7 % Package declaration
8
9 \pkginfograbProvidesExplPackage{intexgral}
10 {
11   name      = {intexgral} ,
12   author    = {Valentin Dao},
13   date      = {2026-01-02} ,
14   creation  = {2025-07-26} ,
15   version   = {3.0.1} ,
16   description = {A LaTeX package for typesetting integrals.}
17 }
18
19 \msg_new:nnn { intexgral } { expl3-too-old }
20 {
21   Your~expl3~installation~is~too~old,~
22   "intexgral"~requires~expl3~dated~2025-05-14~or~later.\iow_newline:
23   Package~loading~has~been~aborted.\iow_newline:
24   \msg_see_documentation_text:n { intexgral }
25 }
26
27 % expl3 version verification
28
29 \@ifpackagelater{expl3}{2025-05-14}{}
30 { \msg_critical:nn { intexgral } { expl3-too-old } }
31
32 % Verifies if \mathbb is defined (default domain style)
33
34 \hook_gput_code:nnn { begindocument/before } { . }
35 { \cs_if_exist:NF \mathbb { \RequirePackage{amsfonts} } }
36
37 % Definition of data types for package options
38
39 \bool_new:N \l__intexgral_invert_limits_bool
40 \bool_new:N \l__intexgral_deactivate_differentials_bool
41 \bool_new:N \l__intexgral_invert_differentials_bool
42 \tl_new:N \l__intexgral_limits_style_tl
43
44 % Package options
45
46 \keys_define:nn { intexgral }
47 {
48   invert-limits .bool_set:N = \l__intexgral_invert_limits_bool,
49   invert-limits .usage:n    = { preamble },
50   invert-limits .initial:n  = { false },
51
52   invert-diff .bool_set:N = \l__intexgral_invert_differentials_bool,
53   invert-diff .usage:n    = { preamble },
54   invert-diff .initial:n  = { false },
```

```

55
56   limits-mode .choice:,
57   limits-mode / limits .code:n =
58     { \tl_set_eq:NN \l__intexgral_limits_style_tl \tex_limits:D },
59   limits-mode / nolimits .code:n =
60     { \tl_set_eq:NN \l__intexgral_limits_style_tl \tex_nolimits:D },
61
62   limits-mode .usage:n    = { preamble },
63   limits-mode .initial:n = { nolimits },
64
65   italic .choice:,
66     italic / true .code:n =
67       { \PassOptionsToPackage{italic=true}{derivative} },
68     italic / false .code:n =
69       { \PassOptionsToPackage{italic=false}{derivative} },
70
71   upright .choice:,
72     upright / true .code:n =
73       { \PassOptionsToPackage{upright=true}{derivative} },
74     upright / false .code:n =
75       { \PassOptionsToPackage{upright=false}{derivative} },
76 }
77
78 % Loads the package options and "derivative" package
79
80 \ProcessKeyOptions[intexgral]
81
82 \RequirePackage{derivative}
83
84 % Warning/Error/Info messages
85
86 \cs_new:Nn \__intexgral_warning_msg_header: {
87   (
88     integral~no.~
89     \int_use:N\g__intexgral_integral_number_int\c_space_tl
90     \msg_line_context:
91   )
92   \iow_newline:
93 }
94
95 \msg_new:nnnn { intexgral } { symb-key-alr-def }
96   { Symbol~key~"\tl_trim_spaces:n{#1}"~is~already~defined. }
97   { Use~\token_to_str:N \RenewSymbolKeyword\ instead. }
98
99 \msg_new:nnnn { intexgral } { symb-key-not-def }
100   { Symbol~key~"\tl_trim_spaces:n{#1}"~is~not~defined. }
101   { Use~\token_to_str:N \NewSymbolKeyword\ instead. }
102
103 \msg_new:nnnn { intexgral } { diff-group-alr-def }
104   { Differential~group~"\tl_trim_spaces:n{#1}"~is~already~defined. }
105   { Use~\token_to_str:N \RenewVarKeyword\ instead. }
106
107 \msg_new:nnnn { intexgral } { diff-group-not-def }
108   { Differential~group~"\tl_trim_spaces:n{#1}"~is~not~defined. }
109   { Use~\token_to_str:N \NewVarKeyword\ instead. }
110
111 \msg_new:nnnn { intexgral } { lim-group-alr-def }
112   { Limits~group~"\tl_trim_spaces:n{#1}"~is~already~defined. }

```

```

113 { Use~\token_to_str:N \RenewLimitsKeyword\ instead. }
114
115 \msg_new:nnnn { intexgral } { lim-group-not-def }
116 { Limits~group~"\tl_trim_spaces:n{#1}"~is~not~defined. }
117 { Use~\NewLimitsKeyword\ instead. }
118
119 \msg_new:nnn { intexgral } { key-exists-for-lim }
120 {
121   Symbol~key~already~defined~with~
122   \token_to_str:N \NewLimitsKeyword\ \__intexgral_warning_msg_header:
123   Key~"#1"~already~exists~for~predefined~limits.~
124   This~can~disrupt~the~functioning~of~the~special~syntax.
125 }
126
127 \msg_new:nnn { intexgral } { unknown-symb }
128 {
129   Unknown~integral~symbol~\__intexgral_warning_msg_header:
130   The~symbol~\tl_trim_spaces:n{#1}~has~been~replaced~with~
131   \token_to_str:N\int.
132 }
133
134 \msg_new:nnn { intexgral } { unequal-dim }
135 {
136   Inconsistent~structure~\__intexgral_warning_msg_header:
137   The~number~of~integrals~(#1),~integrands~(#2),~and~
138   differentials~(#3)~does~not~match.
139 }
140
141 \msg_new:nnn { intexgral } { no-jacobian }
142 {
143   Jacobian~unavailable~\__intexgral_warning_msg_header:
144   A~Jacobian~was~requested~for~the~"#1"~keyword,
145   ~but~none~was~declared~for~the~latter.
146 }
147
148 \msg_new:nnn { intexgral } { diff-not-sup }
149 {
150   Macro~\token_to_str:N\differentials\
151   not~supported~\__intexgral_warning_msg_header:
152   You~can't~use~\token_to_str:N\differentials\
153   with~"nested"~mode~alongside~inverted~differentials.~
154   I~will~simply~ignore~them.
155 }
156
157 \msg_new:nnn { intexgral } { diff-vec-not-supp }
158 {
159   Key~"diff-vec"~not~supported~\__intexgral_warning_msg_header:
160   You~can't~use~"diff-vec"~with~"#1"~mode.~I~will~simply~ignore~this~key.
161 }
162
163 \msg_new:nnn { intexgral } { use-glyph-instead }
164 {
165   Consider~using~the~dedicated~glyph~\__intexgral_warning_msg_header:
166   The~key~"nint"~was~used~with~fewer~than~
167   5~regular~integral~signs.
168 }
169
170 % Argument specifiers variants

```

```

171
172 \cs_generate_variant:Nn \regex_if_match:nnTF { nVTF }
173 \cs_generate_variant:Nn \__intexgral_new_limits_group:nn { nV }
174 \cs_generate_variant:Nn \str_if_eq:nnTF { neTF }
175 \cs_generate_variant:Nn \str_if_eq:nnF { neF }
176 \cs_generate_variant:Nn \keys_if_exist:nnTF { nVTF }
177 \cs_generate_variant:Nn \seq_use:Nn { Ne }
178 \cs_generate_variant:Nn \str_if_eq_p:nn { en }
179
180 % DATA TYPES
181
182 \tl_new:N \l__intexgral_key_name_tl
183 \tl_new:N \l__intexgral_integrand_tl
184 \tl_new:N \l__intexgral_domain_style_tl
185 \tl_new:N \l__intexgral_lower_limit_tl
186 \tl_new:N \l__intexgral_upper_limit_tl
187 \tl_new:N \l__intexgral_integral_symbol_tl
188 \tl_new:N \l__intexgral_differential_options_i_tl
189 \tl_new:N \l__intexgral_differential_options_ii_tl
190 \tl_new:N \l__intexgral_vectorial_differential_style_tl
191 \tl_new:N \l__intexgral_independent_differential_options_tl
192 \tl_new:N \l__intexgral_domain_char_tl
193 \tl_new:N \l__intexgral_domain_dimension_tl
194 \tl_new:N \l__intexgral_default_differential_tl
195 \tl_new:N \l__intexgral_default_vector_differential_tl
196 \tl_const:Nn \c__intexgral_left_bracket_tl { [ }
197 \tl_const:Nn \c__intexgral_right_bracket_tl { ] }
198
199 \tl_set_eq:NN \l__intexgral_integral_symbol_tl \int
200
201 \bool_new:N \l__intexgral_manual_differentials_bool
202 \bool_new:N \l__intexgral_custom_variables_bool
203 \bool_new:N \l__intexgral_separate_integral_bool
204 \bool_new:N \l__intexgral_vectorial_differential_bool
205 \bool_new:N \l__intexgral_jacobian_bool
206 \bool_new:N \l__intexgral_starred_differential_bool
207 \bool_new:N \l__intexgral_has_order_bool
208 \bool_new:N \l__intexgral_has_var_bool
209
210 \bool_set_false:N \l__intexgral_vectorial_differential_bool
211 \bool_set_false:N \l__intexgral_manual_differentials_bool
212 \bool_set_false:N \l__intexgral_custom_variables_bool
213 \bool_set_false:N \l__intexgral_separate_integral_bool
214 \bool_set_false:N \l__intexgral_has_order_bool
215 \bool_set_false:N \l__intexgral_has_var_bool
216
217 \clist_new:N \l__intexgral_variable_i_clist
218 \clist_new:N \l__intexgral_variable_ii_clist
219
220 \seq_new:N \l__intexgral_integrand_seq
221 \seq_new:N \l__intexgral_integral_symbol_seq
222 \seq_new:N \l__intexgral_jacobian_seq
223 \seq_new:N \l__intexgral_differential_seq
224 \seq_new:N \l__intexgral_limits_seq
225 \seq_new:N \l__intexgral_upper_limits_seq
226 \seq_new:N \l__intexgral_lower_limits_seq
227 \seq_new:N \l__intexgral_differential_var_i_seq
228 \seq_new:N \l__intexgral_differential_var_ii_seq

```

```

229 \seq_new:N \l__intexgral_differential_order_i_seq
230 \seq_new:N \l__intexgral_differential_order_ii_seq
231 \seq_new:N \l__intexgral_domain_seq
232
233 \str_new:N \l__intexgral_display_mode_str
234
235 \muskip_new:N \l__intexgral_inline_symbol_muskip
236
237 \prop_new:N \g__intexgral_limits_keyword_prop
238 \prop_new:N \g__intexgral_differential_group_keyword_prop
239
240 \int_new:N \l__intexgral_integral_sequence_int
241 \int_gzero_new:N \g__intexgral_integral_number_int
242
243 % TeX macros adapted to expl3 syntax
244
245 \cs_new_protected:Npn \__intexgral_mkern:N #1
246 { \tex_mkern:D #1 \scan_stop: }
247
248 \cs_new_protected:Npn \__intexgral_mathchoice:nnnn #1#2#3#4
249 { \mathchoice{#1}{#2}{#3}{#4} }
250
251 % SYMBOL
252
253 % General form of the symbol (glyph, limits mode, lower limit, upper limit)
254
255 \cs_new_protected:Npn \__intexgral_parse_integral_symbol:
256 {
257   \tl_use:N \l__intexgral_integral_symbol_tl
258   \tl_use:N \l__intexgral_limits_style_tl
259   \c_math_subscript_token
260   { \l__intexgral_lower_limit_tl }
261   \c_math_superscript_token
262   { \l__intexgral_upper_limit_tl }
263 }
264
265 % Generates the symbols after <limits> is executed
266
267 \cs_new_protected:Nn \__intexgral_generate_integral_sequence:
268 {
269   \int_set:Nn \l__intexgral_integral_sequence_int
270   {
271     \seq_if_empty:NTF \l__intexgral_lower_limits_seq
272     { \seq_count:N \l__intexgral_upper_limits_seq }
273     { \seq_count:N \l__intexgral_lower_limits_seq }
274   }
275   \exp_args:NV
276   \int_step_inline:nn \l__intexgral_integral_sequence_int
277   {
278     \seq_put_right:Nn \l__intexgral_integral_symbol_seq
279     {
280       \__intexgral_set_limits:nn
281       { \seq_item:Nn \l__intexgral_lower_limits_seq { ##1 } }
282       { \seq_item:Nn \l__intexgral_upper_limits_seq { ##1 } }
283       \__intexgral_parse_integral_symbol:
284     }
285   }
286 }

```

```

287
288 % LIMITS
289
290 % Inverts the limit order convention
291
292 \cs_new_protected:Npn \__intexgral_set_limits:nn #1#2
293 {
294   \bool_if:NTF \l__intexgral_invert_limits_bool
295   {
296     \tl_set:Nn \l__intexgral_lower_limit_tl { #2 }
297     \tl_set:Nn \l__intexgral_upper_limit_tl { #1 }
298   }
299   {
300     \tl_set:Nn \l__intexgral_lower_limit_tl { #1 }
301     \tl_set:Nn \l__intexgral_upper_limit_tl { #2 }
302   }
303 }
304
305 % Creates a keyword containing predefined limits
306
307 \cs_new_protected:Npn \__intexgral_new_limits_group:nn #1#2
308 { \prop_put:Nnn \g__intexgral_limits_keyword_prop { #1 } { #2 } }
309
310 % Verifies if the argument corresponds to a valid keyword
311
312 \cs_new:Npn \__intexgral_parse_integral_limit:n #1
313 {
314   \prop_if_in:NnTF \g__intexgral_limits_keyword_prop { #1 }
315   { \prop_item:Nn \g__intexgral_limits_keyword_prop { #1 } }
316   { #1 }
317 }
318
319 % Splits the argument of the key to prepare for parsing
320
321 \cs_new_protected:Npn \__intexgral_parse_limits:n #1
322 {
323   \tl_set:Nn \l_tmpa_tl { #1 }
324   \seq_clear:N \l__intexgral_lower_limits_seq
325   \seq_clear:N \l__intexgral_upper_limits_seq
326   \seq_set_split:NnV \l__intexgral_limits_seq { ; } \l_tmpa_tl
327 }
328
329 % Limits under regular form
330
331 \cs_new_protected:Nn \__intexgral_set_limits_regular:
332 {
333   \seq_map_inline:Nn \l__intexgral_limits_seq
334   {
335     \clist_set:Ne \l_tmpa_clist
336     { \__intexgral_parse_integral_limit:n { ##1 } }
337     \seq_put_right:Ne \l__intexgral_lower_limits_seq
338     { \clist_item:Nn \l_tmpa_clist { 1 } }
339     \seq_put_right:Ne \l__intexgral_upper_limits_seq
340     { \clist_item:Nn \l_tmpa_clist { 2 } }
341   }
342 }
343
344 % Limits under interval form

```

```

345
346 \cs_new_protected:Npn \__intexgral_set_limits_starred:
347 {
348   \seq_map_indexed_inline:Nn \l__intexgral_limits_seq
349   {
350     \clist_set:Nc \l_tmpa_clist
351       { \__intexgral_parse_integral_limit:n { ##2 } }
352     \bool_if:NTF \l__intexgral_invert_limits_bool
353     {
354       \tl_set:Nc \l__intexgral_lower_limit_tl
355       {
356         \clist_item:Nn \l_tmpa_clist { 2 }
357         \tex_mathpunct:D,
358         \clist_item:Nn \l_tmpa_clist { 1 }
359       }
360     }
361     {
362       \tl_set:Nc \l__intexgral_lower_limit_tl
363       {
364         \clist_item:Nn \l_tmpa_clist { 1 }
365         \tex_mathpunct:D,
366         \clist_item:Nn \l_tmpa_clist { 2 }
367       }
368     }
369     \bool_if:NTF \l__intexgral_invert_limits_bool
370     { \seq_put_right:Nc \l__intexgral_upper_limits_seq { }
371       \seq_put_right:Nc \l__intexgral_lower_limits_seq { }
372     }
373     \str_case_e:nnF { \clist_item:Nn \l_tmpa_clist { 1 } }
374     {
375       { -\infty } { \tex_left:D \c__intexgral_right_bracket_tl }
376     }
377     { \tex_left:D \c__intexgral_left_bracket_tl }
378     \tl_use:N \l__intexgral_lower_limit_tl
379     \str_case_e:nnF { \clist_item:Nn \l_tmpa_clist { 2 } }
380     {
381       { +\infty } { \tex_right:D \c__intexgral_left_bracket_tl }
382     }
383     { \tex_right:D \c__intexgral_right_bracket_tl }
384   }
385 }
386 }
387
388 % VARIABLES
389
390 % Creates a keyword containing predefined variables
391
392 \cs_new_protected:Npn \__intexgral_new_differential_group:nnn #1#2#3
393 {
394   \prop_put:Nnn \g__intexgral_differential_group_keyword_prop
395   { #1 } { #2 }
396   \tl_if_blank:nF { #3 }
397   {
398     \clist_set:Nn \l_tmpa_clist { #3 }
399     \seq_set_from_clist:cN
400     { g__intexgral_#1_jacobian_seq }
401     \l_tmpa_clist
402   }

```



```

403 }
404
405 % expl3 equivalent of \odif(*)
406
407 \cs_new:Npn \__intexgral_differentials:nn #1#2
408 {
409   \bool_if:NTF \l__intexgral_starred_differential_bool
410     { \odif*[#1]{#2} }
411     { \odif[#1]{#2} }
412 }
413
414 % Extracts the whole "order" <keyval>
415
416 \cs_new_protected:Npn \__intexgral_extract_differential_order:n #1
417 {
418   \regex_extract_once:nnNT
419     { order\s*=\s*\{?\s*(\d+(?:\s*,\s*\d+)*)\s*\}? }
420     { #1 } \l__intexgral_differential_order_i_seq
421     { \bool_set_true:N \l__intexgral_has_order_bool }
422   \clist_set:Ne \l_tmpa_clist
423     { \seq_item:Nn \l__intexgral_differential_order_i_seq { 2 } }
424   \seq_set_from_clist:NN
425     \l__intexgral_differential_order_ii_seq
426     \l_tmpa_clist
427 }
428
429 % Extracts the whole "var" <keyval>
430
431 \cs_new_protected:Npn \__intexgral_extract_differential_var:n #1
432 {
433   \regex_extract_once:nnNTF
434     { var\s*=\s*(none|all|\{\s*(\d+(?:\s*,\s*\d+)*\s*)\}) }
435     { #1 }
436     \l__intexgral_differential_var_i_seq
437     { \bool_set_true:N \l__intexgral_has_var_bool }
438     {
439       \str_if_in:nnT { #1 } { var }
440       {
441         \seq_put_left:Nn \l__intexgral_differential_var_i_seq { var }
442         \bool_set_true:N \l__intexgral_has_var_bool
443       }
444     }
445   \clist_set:Ne \l_tmpa_clist
446     { \seq_item:Nn \l__intexgral_differential_var_i_seq { 3 } }
447   \seq_set_from_clist:NN
448     \l__intexgral_differential_var_ii_seq
449     \l_tmpa_clist
450 }
451
452 % Extracts all keys other than "order" and "var"
453
454 \cs_new_protected:Npn \__intexgral_remove_differential_order_and_var:n #1
455 {
456   \tl_set:Nn \l__intexgral_independent_differential_options_tl { #1 }
457   \regex_replace_all:nnN
458     {
459       order\s*=\s*\{?\s*\d+(\s*,\s*\d+)*\s*\}?\s*,?
460       |

```

```

461     [^~]var(?:\s*=\s*(?:none|all|\{\s*\d+(?:\s*,\s*\d+)*\s*\}))?,?
462 }
463 { }
464 \l__intexgral_independent_differential_options_tl
465 }
466
467 % Reassembles the variables
468
469 \cs_new_protected:Npn \__intexgral_parse_variable:nn #1#2
470 {
471   \clist_set:Nn \l_tmpa_clist { #2 }
472   \seq_set_from_clist:NN \l_tmpa_seq \l_tmpa_clist
473   \tl_if_empty:nF { #1 }
474   {
475     \__intexgral_extract_differential_order:n { #1 }
476     \__intexgral_extract_differential_var:n { #1 }
477     \__intexgral_remove_differential_order_and_var:n { #1 }
478   }
479   \seq_map_indexed_inline:Nn \l_tmpa_seq {
480     \tl_if_empty:nF { #1 }
481     {
482       \tl_put_right:NV
483         \l__intexgral_differential_options_ii_tl
484         \l__intexgral_independent_differential_options_tl
485       \bool_if:NT \l__intexgral_has_order_bool
486       {
487         \tl_put_right:Ne \l__intexgral_differential_options_ii_tl
488           {
489             ,order=
490             \seq_item:Nn \l__intexgral_differential_order_ii_seq { ##1 }
491           }
492       }
493       \bool_if:NT \l__intexgral_has_var_bool
494       {
495         \int_compare:nNnTF
496           { \seq_count:N \l__intexgral_differential_var_i_seq }
497           =
498           { 1 }
499           {
500             \tl_put_right:Nn
501               \l__intexgral_differential_options_ii_tl
502               { var }
503           }
504           {
505             \str_if_eq:neF { none }
506             { \seq_item:Nn \l__intexgral_differential_var_i_seq { 2 } }
507             {
508               \str_if_eq:neTF { all }
509               { \seq_item:Nn
510                 \l__intexgral_differential_var_i_seq
511                 { 2 }
512               }
513               {
514                 \tl_put_right:Nn
515                   \l__intexgral_differential_options_ii_tl
516                   { var }
517               }
518             }

```

```

519         \seq_if_in:NnT
520         \l__intexgral_differential_var_ii_seq
521         { ##1 }
522         {
523             \seq_pop_left:NN
524             \l__intexgral_differential_var_ii_seq
525             \l_tmpa_tl
526             \tl_put_right:Nn
527             \l__intexgral_differential_options_ii_tl
528             { ,var }
529         }
530     }
531 }
532 }
533 }
534 }
535 \seq_put_right:Ne \l__intexgral_differential_seq
536 {
537     \__intexgral_differentials:nn
538     { \exp_not:V \l__intexgral_differential_options_ii_tl }
539     { \seq_item:Nn \l_tmpa_seq { ##1 } }
540 }
541 \tl_clear:N \l__intexgral_differential_options_ii_tl
542 }
543 }
544
545 % Executes all previous macros to push tokens into the variable sequence
546
547 \cs_new_protected:Npn \__intexgral_parse_differentials:
548 {
549     \bool_if:NTF \l__intexgral_vectorial_differential_bool
550     {
551         \clist_map_inline:Nn \l__intexgral_variable_i_clist
552         {
553             \clist_put_right:Nn \l__intexgral_variable_ii_clist
554             { \l__intexgral_vectorial_differential_style_tl{##1} }
555         }
556         \bool_if:NTF \l__intexgral_custom_variables_bool
557         {
558             \exp_args:NVV \__intexgral_parse_variable:nn
559             \l__intexgral_differential_options_i_tl
560             \l__intexgral_variable_ii_clist
561         }
562         {
563             \exp_args:NV \__intexgral_parse_variable:nn
564             \l__intexgral_differential_options_i_tl
565             {
566                 \l__intexgral_vectorial_differential_style_tl
567                 { \l__intexgral_default_vector_differential_tl }
568             }
569         }
570     }
571     {
572         \bool_if:NTF \l__intexgral_custom_variables_bool
573         {
574             \exp_args:NVV \__intexgral_parse_variable:nn
575             \l__intexgral_differential_options_i_tl
576             \l__intexgral_variable_i_clist

```

```

577     }
578     {
579         \exp_args:NV \__intexgral_parse_variable:nn
580         \l__intexgral_differential_options_i_tl
581         { \l__intexgral_default_differential_tl }
582     }
583 }
584 }
585
586 % SPECIAL SYNTAX
587
588 % Cuts off the value of a key
589
590 \cs_new:Npn \__intexgral_retrieve_key_name:w #1=#2\q_stop { #1 }
591
592 % Extracts first key name of optional argument
593
594 \cs_new_protected:Npn \__intexgral_extract_first_key_name:n #1
595 {
596     \clist_set:Nn \l_tmpa_clist { #1 }
597     \tl_set:Ne \l_tmpa_tl { \clist_item:Nn \l_tmpa_clist { 1 } }
598     \tl_if_in:NnTF \l_tmpa_tl { = }
599     {
600         \tl_set:Ne \l__intexgral_key_name_tl
601         {
602             \exp_last_unbraced:NV
603             \__intexgral_retrieve_key_name:w \l_tmpa_tl \q_stop
604         }
605     }
606     { \tl_set_eq:NN \l__intexgral_key_name_tl \l_tmpa_tl }
607 }
608
609 % Sets the keys as usual, or sets them up manually if special syntax is in use
610
611 \cs_new_protected:Npn \__intexgral_parse_macro_keys:n #1
612 {
613     \keys_if_exist:nVTF { integral } \l__intexgral_key_name_tl
614     { \keys_set:nn { integral } { #1 } }
615     {
616         \regex_split:nnN { : } { #1 } \l_tmpa_seq
617         \str_if_eq:eeTF
618         { \seq_item:Nn \l_tmpa_seq { 2 } }
619         { +j }
620         {
621             \exp_args:NNne \seq_set_item:Nnn \l_tmpa_seq { 2 }
622             { \l__intexgral_default_differential_tl+j }
623         }
624         {
625             \int_compare:nNnT { \seq_count:N \l_tmpa_seq } < { 2 }
626             {
627                 \seq_put_right:NV
628                 \l_tmpa_seq
629                 \l__intexgral_default_differential_tl
630             }
631         }
632         \int_compare:nNnT { \seq_count:N \l_tmpa_seq } < { 3 }
633         { \seq_put_right:Nn \l_tmpa_seq { d } }
634         \seq_set_split:Nne \l_tmpb_seq

```

```

635     { + }
636     { \seq_item:Nn \l_tmpa_seq { 2 } }
637 \keys_set:ne { integral }
638 {
639     limits      = { \seq_item:Nn \l_tmpa_seq { 1 } },
640     variables    = { \seq_item:Nn \l_tmpb_seq { 1 } },
641     mode         =
642     {
643         \str_case:enF { \seq_item:Nn \l_tmpa_seq { 3 } }
644         {
645             { d      } { default }
646             { n      } { nested  }
647             { p      } { product  }
648             { default } { default }
649             { nested } { nested  }
650             { product } { product }
651         }
652         { default }
653     },
654     \bool_if:nT
655     {
656         \int_compare_p:nNn { \seq_count:N \l_tmpb_seq } > { 1 }
657         &&
658         \str_if_eq_p:en { \seq_item:Nn \l_tmpb_seq { 2 } } { j }
659     }
660     { jacobian }
661 }
662 }
663 }
664
665 % PRELIMINARY CONFIGURATIONS
666
667 % Verifies that <limits> and <variables> (and <integrand>) have the same size
668
669 \cs_new_protected:Nn \__intexgral_check_sequence_size:
670 {
671     \int_compare:nNnTF
672     { \seq_count:N \l__intexgral_integral_symbol_seq }
673     =
674     { \seq_count:N \l__intexgral_integrand_seq }
675     {
676         \int_compare:nNnF
677         { \seq_count:N \l__intexgral_integrand_seq }
678         =
679         { \seq_count:N \l__intexgral_differential_seq }
680         {
681             \msg_warning:nneee { intexgral } { unequal-dim }
682             { \seq_count:N \l__intexgral_integral_symbol_seq }
683             { \seq_count:N \l__intexgral_integrand_seq }
684             { \seq_count:N \l__intexgral_differential_seq }
685         }
686     }
687     {
688         \msg_warning:nneee { intexgral } { unequal-dim }
689         { \seq_count:N \l__intexgral_integral_symbol_seq }
690         { \seq_count:N \l__intexgral_integrand_seq }
691         { \seq_count:N \l__intexgral_differential_seq }
692     }

```

```

693 }
694
695 % Performs preparatory actions before executing the main macro
696
697 \cs_new_protected:Nn \__intexgral_integral_preconfiguration:
698 {
699   \int_gincr:N \g__intexgral_integral_number_int
700
701   \bool_if:NT \l__intexgral_separate_integral_bool
702   {
703     \seq_set_split:NnV
704       \l__intexgral_integrand_seq
705       { ; }
706       \l__intexgral_integrand_tl
707   }
708
709   \seq_if_empty:NT \l__intexgral_integral_symbol_seq
710   {
711     \seq_put_right:Nn \l__intexgral_integral_symbol_seq
712       { \__intexgral_parse_integral_symbol: }
713   }
714
715   \seq_if_empty:NT \l__intexgral_integrand_seq
716   {
717     \seq_put_right:NV
718       \l__intexgral_integrand_seq
719       \l__intexgral_integrand_tl
720   }
721
722   \bool_if:NF \l__intexgral_deactivate_differentials_bool
723   { \__intexgral_parse_differentials: }
724
725   \regex_if_match:nVTF { \c{differentials} } \l__intexgral_integrand_tl
726   { \bool_set_true:N \l__intexgral_manual_differentials_bool }
727   { \bool_set_false:N \l__intexgral_manual_differentials_bool }
728
729   \bool_if:NT \l__intexgral_separate_integral_bool
730   { \__intexgral_check_sequence_size: }
731 }
732
733 % Default mode
734
735 \cs_new_protected:Nn \__intexgral_print_default_integral:
736 {
737   \bool_if:NT \l__intexgral_manual_differentials_bool
738   {
739     \cs_set_protected:Npn \differentials
740       { \seq_use:Nn \l__intexgral_differential_seq { } }
741   }
742   \seq_use:Nn \l__intexgral_integral_symbol_seq
743     { \__intexgral_mkern:N \l__intexgral_inline_symbol_muskip }
744   \bool_if:nT
745   {
746     \l__intexgral_invert_differentials_bool
747     &&
748     !\l__intexgral_manual_differentials_bool
749   }
750   {

```

```

751     \seq_use:Nn \l__intexgral_differential_seq { } \tex_mathop:D{} \!
752     \bool_if:NT \l__intexgral_vectorial_differential_bool { \cdot \: }
753   }
754   \seq_use:Nn \l__intexgral_integrand_seq { }
755   \bool_if:NT \l__intexgral_jacobian_bool
756     { \seq_use:Nn \l__intexgral_jacobian_seq { } }
757   \bool_if:nT
758     {
759       !\l__intexgral_invert_differentials_bool
760       &&
761       !\l__intexgral_manual_differentials_bool
762     }
763     {
764       \bool_if:NT \l__intexgral_vectorial_differential_bool { \cdot }
765       \seq_use:Nn \l__intexgral_differential_seq { }
766     }
767   }
768
769   % Nested mode
770
771   \cs_new_protected:Nn \__intexgral_print_nested_integral:
772   {
773     \bool_if:NT \l__intexgral_manual_differentials_bool
774     {
775       \cs_set_protected:Npn \differentials
776       { \seq_use:Nn \l__intexgral_differential_seq { } }
777     }
778     \bool_if:NT \l__intexgral_vectorial_differential_bool
779     { \msg_warning:nnn { intexgral } { diff-vec-not-supp } { nested } }
780     \int_step_inline:nn { \seq_count:N \l__intexgral_integral_symbol_seq }
781     {
782       \seq_item:Nn \l__intexgral_integral_symbol_seq { ##1 }
783       \bool_if:NTF \l__intexgral_invert_differentials_bool
784       {
785         \bool_if:NT \l__intexgral_manual_differentials_bool
786         {
787           \msg_warning:nn { intexgral } { diff-not-sup }
788           \cs_set_eq:NN \differentials \scan_stop:
789         }
790         \seq_item:Nn \l__intexgral_differential_seq { ##1 }
791         \tex_mathop:D{} \!
792       }
793       {
794         \seq_item:Nn \l__intexgral_integrand_seq { ##1 }
795         \bool_if:NT \l__intexgral_jacobian_bool
796         { \seq_item:Nn \l__intexgral_jacobian_seq { ##1 } }
797       }
798     }
799     \bool_if:NTF \l__intexgral_invert_differentials_bool
800     {
801       \seq_use:Nn \l__intexgral_integrand_seq {}
802       \bool_if:NT \l__intexgral_jacobian_bool
803       { \seq_use:Nn \l__intexgral_jacobian_seq { } }
804     }
805     {
806       \bool_if:NF \l__intexgral_manual_differentials_bool
807       {
808         \seq_use:Nn \l__intexgral_differential_seq {}

```

```

809     }
810   }
811 }
812
813 % Product mode
814
815 \cs_new_protected:Nn \__intexgral_print_product_integral:
816 {
817   \bool_if:NT \l__intexgral_manual_differentials_bool
818   {
819     \cs_set_protected:Npn \differentials
820     {
821       \seq_gpop_left:NN \l_tmpa_tl
822       \tl_use:N \l_tmpa_tl
823     }
824   }
825   \bool_if:NT \l__intexgral_vectorial_differential_bool
826   { \msg_warning:nnn { intexgral } { diff-vec-not-supp } { product } }
827   \int_step_inline:nn { \seq_count:N \l__intexgral_integral_symbol_seq }
828   {
829     \seq_item:Nn \l__intexgral_integral_symbol_seq { ##1 }
830     \bool_if:nT
831     {
832       \l__intexgral_invert_differentials_bool
833       &&
834       !\l__intexgral_manual_differentials_bool
835     }
836     {
837       \seq_item:Nn
838       \l__intexgral_differential_seq { ##1 }
839       \tex_mathop:D{} \!
840     }
841     \seq_item:Nn \l__intexgral_integrand_seq { ##1 }
842     \bool_if:NT \l__intexgral_jacobian_bool
843     { \seq_item:Nn \l__intexgral_jacobian_seq { ##1 } }
844     \bool_if:nT
845     {
846       !\l__intexgral_invert_differentials_bool
847       &&
848       !\l__intexgral_manual_differentials_bool
849     }
850     { \seq_item:Nn \l__intexgral_differential_seq { ##1 } }
851   }
852 }
853
854 % General macro for typesetting
855
856 \cs_new_protected:Nn \__intexgral_print_integral:
857 {
858   \__intexgral_integral_preconfiguration:
859   \str_case:VnF \l__intexgral_display_mode_str
860   {
861     { default } { \__intexgral_print_default_integral: }
862     { nested } { \__intexgral_print_nested_integral: }
863     { product } { \__intexgral_print_product_integral: }
864   }
865   { \__intexgral_print_default_integral: }
866   \seq_gclear:N \l__intexgral_differential_seq

```



```

867 }
868
869 % KEYS
870
871 \keys_define:nn { integral } {
872
873   % LIMITS
874
875   mode .choice:,
876     mode / default .code:n =
877     {
878       \bool_set_false:N \l__intexgral_separate_integral_bool
879       \str_set:Nn \l__intexgral_display_mode_str { default }
880     },
881     mode / nested .code:n =
882     {
883       \bool_set_true:N \l__intexgral_separate_integral_bool
884       \str_set:Nn \l__intexgral_display_mode_str { nested }
885     },
886     mode / product .code:n =
887     {
888       \bool_set_true:N \l__intexgral_separate_integral_bool
889       \str_set:Nn \l__intexgral_display_mode_str { product }
890     },
891     mode .default:n = { default },
892
893   limits .code:n =
894   {
895     \__intexgral_parse_limits:n { #1 }
896     \__intexgral_set_limits_regular:
897     \__intexgral_generate_integral_sequence:
898   },
899
900   limits* .code:n =
901   {
902     \__intexgral_parse_limits:n { #1 }
903     \__intexgral_set_limits_starred:
904     \__intexgral_generate_integral_sequence:
905   },
906
907   llimit .tl_set:N = \l__intexgral_lower_limit_tl,
908
909   ulimit .tl_set:N = \l__intexgral_upper_limit_tl,
910
911   symbol .code:n =
912   {
913     \cs_if_exist:NTF #1
914     { \tl_set_eq:NN \l__intexgral_integral_symbol_tl #1 }
915     {
916       \msg_warning:nnn { intexgral } { unknown-symb } { #1 }
917       \tl_set_eq:NN \l__intexgral_integral_symbol_tl \int
918     }
919   },
920
921   % SYMBOL SHORTCUT
922
923   nint .code:n =
924   {

```

```

925 \int_compare:nNnT { #1 } < { 5 }
926 { \msg_warning:nn { intexgral } { use-glyph-instead } }
927
928 \tl_clear:N \l__intexgral_integral_symbol_tl
929
930 \int_step_inline:nn { #1 }
931 {
932   \tl_put_right:Nn \l__intexgral_integral_symbol_tl
933     { \int }
934   \int_compare:nNnT { ##1 } < { #1 }
935   {
936     \tl_put_right:Nn \l__intexgral_integral_symbol_tl
937       {
938         \__intexgral_mathchoice:nnnn
939         { \tex_mkern:D -12mu \scan_stop: }
940         { \tex_mkern:D -8mu \scan_stop: }
941         { \tex_mkern:D -4mu \scan_stop: }
942         { \tex_mkern:D -2mu \scan_stop: }
943       }
944     }
945   }
946 },
947
948 % LIMITS SHORTCUTS
949
950 domain .code:n =
951 {
952   \tl_set:Nn \l_tmpa_tl { #1 }
953   \seq_set_split:NnV \l__intexgral_domain_seq { * } \l_tmpa_tl
954   \seq_map_inline:Nn \l__intexgral_domain_seq
955   {
956     \tl_if_empty:NF \l__intexgral_lower_limit_tl
957     { \tl_put_right:Nn \l__intexgral_lower_limit_tl { \times } }
958     \tl_set:Ne \l__intexgral_domain_char_tl
959     { \exp_args:Ne \str_uppercase:n { \tl_head:n { ##1 } } }
960     \tl_set:Ne \l__intexgral_domain_dimension_tl
961     { \tl_tail:n { ##1 } }
962     \tl_put_right:Ne \l__intexgral_lower_limit_tl
963     {
964       \exp_not:N \l__intexgral_domain_style_tl
965       { \l__intexgral_domain_char_tl }
966       \c_math_superscript_token { \l__intexgral_domain_dimension_tl }
967     }
968   }
969 },
970
971 domain* .code:n =
972 {
973   \tl_set:Nn \l_tmpa_tl { #1 }
974   \seq_set_split:NnV \l__intexgral_domain_seq { * } \l_tmpa_tl
975   \seq_map_inline:Nn \l__intexgral_domain_seq
976   {
977     \tl_if_empty:NF \l__intexgral_lower_limit_tl
978     { \tl_put_right:Nn \l__intexgral_lower_limit_tl { \times } }
979     \tl_set:Ne \l__intexgral_domain_char_tl
980     { \exp_args:Ne \str_uppercase:n { \tl_head:n { ##1 } } }
981     \tl_set:Ne \l__intexgral_domain_dimension_tl
982     { \tl_tail:n { ##1 } }

```

```

983         \tl_put_right:N \l__intexgral_lower_limit_tl
984         {
985             \exp_not:N \l__intexgral_domain_style_tl
986             { \l__intexgral_domain_char_tl }
987             \c_math_subscript_token { \l__intexgral_domain_dimension_tl }
988         }
989     }
990 },
991
992 boundary .code:n =
993     { \tl_set:Nn \l__intexgral_lower_limit_tl { \partial #1 } },
994
995 % VARIABLES
996
997 variables .code:n =
998     {
999         \bool_set_true:N \l__intexgral_custom_variables_bool
1000         \str_if_eq:nnTF { #1 } { none }
1001         { \bool_set_true:N \l__intexgral_deactivate_differentials_bool }
1002         {
1003             \prop_get:NnNTF
1004             \g__intexgral_differential_group_keyword_prop { #1 } \l_tmpa_tl
1005             {
1006                 \clist_set:NV \l__intexgral_variable_i_clist \l_tmpa_tl
1007                 \seq_if_exist:cTF { g__intexgral_#1_jacobian_seq }
1008                 {
1009                     \seq_set_eq:Nc
1010                     \l__intexgral_jacobian_seq
1011                     { g__intexgral_#1_jacobian_seq }
1012                 }
1013                 { \msg_warning:nnn { intexgral } { no-jacobian } { #1 } }
1014             }
1015             { \clist_set:Nn \l__intexgral_variable_i_clist { #1 } }
1016         }
1017     },
1018
1019 jacobian .code:n =
1020     { \bool_set_true:N \l__intexgral_jacobian_bool },
1021 diff-vec .code:n =
1022     { \bool_set_true:N \l__intexgral_vectorial_differential_bool },
1023 diff-star .code:n =
1024     { \bool_set_true:N \l__intexgral_starred_differential_bool },
1025
1026 diff-symb .code:n =
1027     {
1028         \cs_set:Npn \__intexgral_differentials:nn ##1##2 {
1029             \bool_if:NTF \l__intexgral_starred_differential_bool
1030             { #1*{##1}{##2} }
1031             { #1[##1]{##2} }
1032         }
1033     },
1034
1035 diff-options .tl_set:N = \l__intexgral_differential_options_i_tl,
1036 }
1037
1038 \keys_define:nn { IntegralSetup }
1039 {
1040     defaultvar .tl_set:N = \l__intexgral_default_differential_tl,

```

```

1041     defaultvar* .tl_set:N = \l__intexgral_default_vector_differential_tl,
1042     vectorstyle .tl_set:N = \l__intexgral_vectorial_differential_style_tl,
1043     domainstyle .tl_set:N = \l__intexgral_domain_style_tl,
1044     symbolskip .tl_set:N = \l__intexgral_inline_symbol_muskip,
1045     hide-diff .bool_set:N = \l__intexgral_deactivate_differentials_bool,
1046   }
1047
1048 % END USER MACROS
1049
1050 % Limits keywords
1051
1052 \NewDocumentCommand\NewLimitsKeyword{ m m }
1053 {
1054   \prop_get:NnNTF \g__intexgral_limits_keyword_prop { #1 } \l_tmpa_tl
1055   { \msg_error:nnn { intexgral } { lim-group-alr-def } { #1 } }
1056   {
1057     \bool_if:NTF \l__intexgral_invert_limits_bool
1058     {
1059       \clist_set:Nn \l_tmpa_clist { #2 }
1060       \clist_reverse:N \l_tmpa_clist
1061       \__intexgral_new_limits_group:nV { #1 } \l_tmpa_clist
1062     }
1063     { \__intexgral_new_limits_group:nn { #1 } { #2 } }
1064   }
1065 }
1066
1067 \NewDocumentCommand\RenewLimitsKeyword{ m m }
1068 {
1069   \prop_pop:NnNTF \g__intexgral_limits_keyword_prop { #1 } \l_tmpa_tl
1070   {
1071     \bool_if:NTF \l__intexgral_invert_limits_bool
1072     {
1073       \clist_set:Nn \l_tmpa_clist { #2 }
1074       \clist_reverse:N \l_tmpa_clist
1075       \__intexgral_new_limits_group:nV { #1 } \l_tmpa_clist
1076     }
1077     { \__intexgral_new_limits_group:nn { #1 } { #2 } }
1078   }
1079   { \msg_error:nnn { intexgral } { lim-group-not-def } }
1080 }
1081
1082 \NewDocumentCommand\ProvideLimitsKeyword{ m m }
1083 {
1084   \prop_get:NnNF \g__intexgral_limits_keyword_prop { #1 } \l_tmpa_tl
1085   {
1086     \bool_if:NTF \l__intexgral_invert_limits_bool
1087     {
1088       \clist_set:Nn \l_tmpa_clist { #2 }
1089       \clist_reverse:N \l_tmpa_clist
1090       \__intexgral_new_limits_group:nV { #1 } \l_tmpa_clist
1091     }
1092     { \__intexgral_new_limits_group:nn { #1 } { #2 } }
1093   }
1094 }
1095
1096 \NewDocumentCommand\DeclareLimitsKeyword{ m m }
1097 {
1098   \prop_pop:NnNT \g__intexgral_limits_keyword_prop { #1 } \l_tmpa_tl

```

```

1099     {
1100         \bool_if:NTF \l__intexgral_invert_limits_bool
1101         {
1102             \clist_set:Nn \l_tmpa_clist { #2 }
1103             \clist_reverse:N \l_tmpa_clist
1104             \__intexgral_new_limits_group:nV { #1 } \l_tmpa_clist
1105         }
1106         { \__intexgral_new_limits_group:nn { #1 } { #2 } }
1107     }
1108 }
1109
1110 % Variable keywords
1111
1112 \NewDocumentCommand\NewVariableKeyword{ m m o }
1113 {
1114     \prop_get:NnNTF \g__intexgral_differential_group_keyword_prop { #1 }
1115     \l_tmpa_tl
1116     { \msg_error:nnn { intexgral } { diff-group-alr-def } { #1 } }
1117     { \__intexgral_new_differential_group:nnn { #1 } { #2 } { #3 } }
1118 }
1119
1120 \NewDocumentCommand\RenewVariableKeyword{ m m o }
1121 {
1122     \prop_pop:NnNTF \g__intexgral_differential_group_keyword_prop { #1 }
1123     \l_tmpa_tl
1124     {
1125         \seq_clear:c { g__intexgral_#1_jacobian_seq }
1126         \__intexgral_new_differential_group:nnn { #1 } { #2 } { #3 }
1127     }
1128     { \msg_error:nnn { intexgral } { diff-group-not-def } { #1 } }
1129 }
1130
1131 \NewDocumentCommand\ProvideVariableKeyword{ m m o }
1132 {
1133     \prop_get:NnNF \g__intexgral_differential_group_keyword_prop { #1 }
1134     \l_tmpa_tl
1135     { \__intexgral_new_differential_group:nnn { #1 } { #2 } { #3 } }
1136 }
1137
1138 \NewDocumentCommand\DeclareVariableKeyword{ m m o }
1139 {
1140     \prop_pop:NnNT \g__intexgral_differential_group_keyword_prop { #1 }
1141     \l_tmpa_tl
1142     {
1143         \seq_clear:c { g__intexgral_#1_jacobian_seq }
1144         \__intexgral_new_differential_group:nnn { #1 } { #2 } { #3 }
1145     }
1146 }
1147
1148 % Symbol keys
1149
1150 \NewDocumentCommand\NewSymbolKeyword{ m m }
1151 {
1152     \prop_if_in:NnT \g__intexgral_limits_keyword_prop { #1 }
1153     { \msg_warning:nnn { intexgral } { key-exists-for-lim } { #1 } }
1154     \keys_if_exist:nnTF { integral } { #1 }
1155     { \msg_error:nnn { intexgral } { symb-key-alr-def } { #1 } }
1156     {

```

```

1157     \keys_define:nn { integral }
1158     {
1159         #1 .meta:n =
1160         {
1161             symbol=#2,
1162             llimit=##1
1163         },
1164     }
1165 }
1166 }
1167
1168 \NewDocumentCommand\RenewSymbolKeyword{ m m }
1169 {
1170     \prop_if_in:NnT \g__intexgral_limits_keyword_prop { #1 }
1171     { \msg_warning:nnn { intexgral } { key-exists-for-lim } { #1 } }
1172     \keys_if_exist:nnTF { integral } { #1 }
1173     {
1174         \keys_define:nn { integral }
1175         {
1176             #1 .undefine:
1177             #1 .meta:n =
1178             {
1179                 symbol=#2,
1180                 llimit=##1
1181             }
1182         }
1183     }
1184     { \msg_error:nnn { intexgral } { symb-key-not-def } }
1185 }
1186
1187 \NewDocumentCommand\ProvideSymbolKeyword{ m m }
1188 {
1189     \prop_if_in:NnT \g__intexgral_limits_keyword_prop { #1 }
1190     { \msg_warning:nnn { intexgral } { key-exists-for-lim } { #1 } }
1191     \keys_if_exist:nnF { integral } { #1 }
1192     {
1193         \keys_define:nn { integral }
1194         {
1195             #1 .meta:n =
1196             {
1197                 symbol=#2,
1198                 llimit=##1
1199             },
1200         }
1201     }
1202 }
1203
1204 \NewDocumentCommand\DeclareSymbolKeyword{ m m }
1205 {
1206     \keys_define:nn { integral }
1207     {
1208         #1 .undefine:
1209         #1 .meta:n =
1210         {
1211             symbol=#2,
1212             llimit=##1
1213         },
1214     }

```

```

1215 }
1216
1217 \NewDocumentCommand\IntegralSetup{ m }
1218 { \keys_set:nn { IntegralSetup } { #1 } }
1219
1220 \NewDocumentCommand\integral{ 0{} m }
1221 {
1222   \group_begin:
1223   \tl_if_empty:nF { #1 }
1224   {
1225     \__intexgral_extract_first_key_name:n { #1 }
1226     \__intexgral_parse_macro_keys:n { #1 }
1227   }
1228   \tl_set:Nn \l__intexgral_integrand_tl { #2 }
1229   \__intexgral_print_integral:
1230   \group_end:
1231 }
1232
1233 % Preamble only
1234
1235 \@onlypreamble\intexgralsetup
1236
1237 % Included keywords
1238
1239 % Symbol keywords
1240 \NewSymbolKeyword{single}      {\int}
1241 \NewSymbolKeyword{double}     {\iint}
1242 \NewSymbolKeyword{triple}     {\iiint}
1243 \NewSymbolKeyword{quadruple}  {\iiiiint}
1244 \NewSymbolKeyword{contour}    {\oint}
1245 \NewSymbolKeyword{surface}    {\oiint}
1246 \NewSymbolKeyword{volume}     {\oiint}
1247
1248 % Limits keywords
1249 \NewLimitsKeyword{ab}         {a, b}
1250 \NewLimitsKeyword{real}       {-\infty, +\infty}
1251 \NewLimitsKeyword{positive}   {0, +\infty}
1252 \NewLimitsKeyword{negative}   {-\infty, 0}
1253 \NewLimitsKeyword{unit}       {0, 1}
1254 \NewLimitsKeyword{circle}     {0, 2\pi}
1255 \NewLimitsKeyword{scircle}    {0, \pi}
1256 \NewLimitsKeyword{qcircle}    {0, \frac{\pi}{2}}
1257 \NewLimitsKeyword{height}     {0, H}
1258 \NewLimitsKeyword{radius}     {0, R}
1259 \NewLimitsKeyword{length}     {0, L}
1260 \NewLimitsKeyword{time}       {0, T}
1261
1262 % Variable keywords
1263 \NewVariableKeyword{cartesian} {x, y, z}
1264 \NewVariableKeyword{planar}    {x, y}
1265 \NewVariableKeyword{polar}     {r, \theta} [r]
1266 \NewVariableKeyword{cylindrical} {r, \theta, z} [r]
1267 \NewVariableKeyword{spherical} {r, \theta, \phi} [r^2, \sin\theta]
1268
1269 % Default parameters
1270
1271 \IntegralSetup
1272 {

```

```

1273     defaultvar = {x},
1274     defaultvar* = {r},
1275     vectorstyle = \vec,
1276     domainstyle = \mathbb,
1277     symbolskip = {-6mu},
1278     hide-diff = false
1279 }
1280
1281 \ExplSyntaxOff

```