intexgral package*

Valentin Dao†

Released 2025-07-26

Abstract

While integral composition is very common in LaTeX, it often proves to be impractical. When the expression becomes complex, it becomes difficult to modify its various elements in a source an illegible source code. To address this problem, the intexgral package offers central macro whose only argument is the integrand. Everything else (limits, differentials...) can be modified using a ⟨keyval⟩ interface. Unlike the traditional method, where the user choses the limits 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 LaTeX3 group of packages l3kernel and l3package. In addition, intexgral uses the derivative package to facilitate the manipulation of differentials.

Contents

---

1 Package options

These options can be declared woth with the following syntax when the package is loaded in the preamble:

$$\text{\textbackslash usepackage[}\langle\textit{keyval}\rangle\text{]\{intexgral\}}$$

---
invert-limits $\langle\textit{boolean}\rangle$ (default: false)

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

---
invert-differentials $\langle\textit{boolean}\rangle$ (default: false)

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

---
hide-differentials $\langle\textit{boolean}\rangle$ (default: false)

If the user wishes to hide the differentials throughout the whole document, this key completely deactivates their automatic composition.

---
italic $\langle\textit{boolean}\rangle$ (default: false)
upright $\langle\textit{boolean}\rangle$ (default: true)

These are the two keys of the derivative package.

2 Presentation of the main macro

\integral [$\langle\textit{keyval list}\rangle$]{$\langle\textit{integrand}\rangle$}

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

```
\integral{f(x)}
```

$$\int f(x)\,\mathrm{d}x \tag{1}$$

2

As this macro revolves around the use of keys, the first part of this documentation will present them grouped by themes. The second part will document the auxiliary macros that complement the functionality of certain keys.

3 List of keys

3.1 Integration limits

3.1.1 Defining the limits

`limits` {⟨*pseudo-clist*⟩}, ⟨*keyword*⟩
`limits*` {⟨*pseudo-clist*⟩}, ⟨*keyword*⟩

This key determines the integration limits to be used, following the convention explained in the abstract.

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

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

The starred version of the key expresses the limits as an interval. The direction of the brackets automatically adjusts to the presence of $\pm\infty$.

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

$$\int_{[1,10]} f(x)\,\mathrm{d}x \tag{3}$$

The major strength of `limits` is that it allows ot specify the limits of several integrals and composes them for you. To do this, thge limits must be separated by semicolns to allow the package to correctly evaluate to which integral they belong to.

```
\integral[
   limits={0, R; 0, H; 0, 2\pi},
   variable={\rho, z, \phi}
   ]{\rho z \phi}
```

$$\int_0^R \int_0^H \int_0^{2\pi} \rho z\phi\,\mathrm{d}\rho\,\mathrm{d}z\,\mathrm{d}\phi \tag{4}$$

3.1.2   Separate the integral

⟨*boolean*⟩ (default: false)

This key separates the different variables of the intragral and isolates them. It is therefore necessary to use it *before* limits so that the latter knows in which mode to operate in. Morevoer, this key requires that we indicate how to separate the integrand. This additional step can be done using the same logic as for the limits, i.e. using a semicolon.

```
\integral[
  int-split=true,
  limits={radius; height; circle},
  variable={\rho, z, \phi}
]{\rho; z; \phi}
```

$$\int_0^R \rho \, \mathrm{d}\rho \int_0^H z \, \mathrm{d}z \int_0^{2\pi} \phi \, \mathrm{d}\phi \tag{5}$$

To function correctly, it is obvious that the keys limits and variable[1], as well as the integrand, have to have the same dimension. If it is not the case, the compilation won't fail, but the integral expression might no longer make any sense. Numerous warning messages will be there to inform you of this and tell you what is wrong. So don't forget the commas and/or semicolons, even if an element remains empty!

3.1.3   Chosing the limits' positioning

limits, nolimits (default: nolimits)

Equivalent to using \limits or \nolimits.

```
\integral[
  limits-mode=limits,
  limits={a, b}
]{f(x)}
```

$$\int_a^b f(x) \, \mathrm{d}x \tag{6}$$

3.2   Symbol (and more about limits)

The keys limits and limits* are very useful when the final expression of the integral is determined (in other words, when the limits of each variable have been defined). However, this only covers... the final expressino! For more general cases, they are relativaly inconvenient. To typeset a double integral over a surface S, one would have to write limits={,;,S}. Needless to say, it is rather unsuitable for the user and not optimal for the package. Furthermore, the glyph won't be correct. The set of forthcoming keys therefore offers an easy way to modify the integral symbol and limits.

---

[1]To be discovered.

3.2.1 Selecting a symbol

`int-symb` ⟨*control sequence*⟩

This key accepts a macro designating an integral symbol. For instance, `int-symb=\oint` is equivalent to using `\oint_{...}^{...}`. By default, the intexgral package tries to load all symbols from unicode-math[2], as it offers the largest variety of them. If a symbol is not defined, a warning message will be issued and the missing macro will be substituted to `\int`.

```
\integral[
   int-symb=\oint,
   lower-lim=\mathcal{C},
   diff-vec=true,
   ]{f(\vec{r})}
```

$$\oint_{\mathcal{C}} f(\vec{r}) \cdot \mathrm{d}\vec{r} \tag{7}$$

3.2.2 Generating many integrals

`nint` ⟨*integer*⟩

This key accepts an integer n that allows to typeset n integrals. It is recommended to use this key only if the number exceeds 4.

```
\integral[
   nint=5,
   lower-lim=\Omega,
   variable={x_1, x_2, x_3, x_4, x_5}
   ]{f(x_1, x_2, x_3, x_4, x_5)}
```

$$\idotsint_{\Omega} f(x_1, x_2, x_3, x_4, x_5)\, \mathrm{d}x_1\, \mathrm{d}x_2\, \mathrm{d}x_3\, \mathrm{d}x_4\, \mathrm{d}x_5 \tag{8}$$

3.2.3 Defining limits (when there is only one symbol)

`lower-lim` {⟨*lower limit*⟩}
`upper-lim` {⟨*upper limit*⟩}

These two keys allow to specify the lower and upper limits respectively. If you need to specify both limits, the key `limits` should be used instead.

```
\integral[
   lower-lim={x^2 + y^2 \leq 1},
   variable={x, y}
   ]{f(x, y)}
```

---

[2]See documentation for the complete list of symbols.

$$\int_{x^2+y^2\leq 1} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{9}$$

The previous keys (excluding `nint`) cover a fairly — I hope — general use. Nonetheless, further optimisation is possible to save even more time. That is why in addition of `int-symb`, `lower-lim` and `upper-lim`, intexgral offers *shortcut* keys:

3.2.4 Combining symbol and limit

`single`	`{⟨lower limit⟩}`	(symbol: `\int`)
`double`	`{⟨lower limit⟩}`	(symbol: `\iint`)
`triple`	`{⟨lower limit⟩}`	(symbol: `\iiint`)
`quadruple`	`{⟨lower limit⟩}`	(symbol: `\iiiint`)
`contour`	`{⟨lower limit⟩}`	(symbol: `\oint`)
`surface`	`{⟨lower limit⟩}`	(symbol: `\oiint`)
`volume`	`{⟨lower limit⟩}`	(symbol: `\oiiint`)

All these keys allow the action of `int-symb` and `lower-lim` to be grouped together, making it easier to chose the symbol and the lower bound at the same time. The value of the key is optional, in which case the symbol will change but the bound will remain empty.

```
\integral[
  double=S,
  variable={x, y}
  ]{xy}
```

$$\iint_S xy\,\mathrm{d}x\,\mathrm{d}y \tag{10}$$

3.2.5 Recurring limits motif

As with symbols, lower bounds sometimes follow common patterns that can be generalised with keys, avoiding overloading the expression of `lower-lim`.

`domain` `{⟨*-list⟩}`

This key accepts a list delimited by asterisks. Then, each element of the list will be parsed as follows:

- The first token of the item is passed to `\mathbb` (preceded by `\uppercase` in case the SHIFT key is too far away for your fingers).

- The remaining tokens — which may be empty — are interpreted as the Cartesian power of the domain.

```
\integral[
  double,
  domain={r*r},
  variable={x, y}
  ]{xy}
```

$$\iint_{\mathbb{R}\times\mathbb{R}} xy \, \mathrm{d}x \, \mathrm{d}y \qquad (11)$$

---

boundary {⟨*lower limit*⟩}

This key simply placed the ∂ symbol before the lower limit.

```
\integral[
   contour,
   variable=r,
   diff-vec=true,
   boundary=S
   ]{G(\vec{r})}
```

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

3.3   Differentials

3.3.1   Specifying the variables

---

variable {⟨*clist*⟩}, ⟨*keyword*⟩, none

This key allows to specify the integration variables as a comma list. If this key is not specified, the package will display $\mathrm{d}x$ by default (or $\mathrm{d}\vec{r}$ if diff-vec is enabled). If none is parsed, no differentials will be shown.

```
\integral[
   triple=V,
   variable={x, y, z}
   ]{f(x, y, z)}
```

$$\iiint_V f(x,y,z) \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z \qquad (13)$$

3.3.2   Including the Jacobian

---

jacobien ⟨*boolean*⟩ (default: false)

This key enables/disables the display of the Jacobian when defined by \NewDifferentialKeyword (see)

```
V_\textrm{sphere} =
\integral[
   int-split=true,
   limits={0, R; 0, 2\pi; 0, \pi},
   variable=spherical,
   jacobian=true,
   ]{;;}
```

$$V_{\text{sphere}} = \int_0^R r^2 \, \mathrm{d}r \int_0^{2\pi} \sin\theta \, \mathrm{d}\theta \int_0^{\pi} \mathrm{d}\phi \tag{14}$$

3.3.3 Modifying the symbol

`diff-symb` ⟨*control sequence*⟩

As mentioned earlier, the derivative package is used to format differentials. This key allows to change the style of the differentials among those defined by \NewDifferential

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

$$\int e^{+\dfrac{\imath S[q(t)]}{\hbar}} \, \mathcal{D}q(t) \tag{15}$$

3.3.4 Opting for the starred variant

`diff-star` ⟨*boolean*⟩ (default: false)

If this key is set to true, it is equivalent to using \odif* or its variants.

```
\integral[
  double,
  variable={x, y},
  diff-star=true
  ]{x^2 \exp(y)}
```

$$\iint x^2 \exp(y) \, \mathrm{d}_{x,y} \tag{16}$$

3.3.5 Designating options

`diff-options` {⟨*keyval list*⟩}

This key accepts a list of keys as it would have been written as an optionnal 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]{...}.

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

$$\int \bar{\psi}(x)(\imath\gamma^\mu\partial_\mu - m)\psi(x) \, \mathrm{d}^4 x \tag{17}$$

3.3.6   Creating a path integral

---

diff-vec ⟨*boolean*⟩

This key applies \vec to each integration variable and places \cdot as a seperator.

```
\integral[
  double=S,
  variable=S,
  diff-vec=true,
  ]{\vec{F}}
```

$$\iint_S \vec{F} \cdot \mathrm{d}\vec{S} \tag{18}$$

3.3.7   Customising the vector

---

diff-vec-style ⟨*control sequence*⟩

There are many different vector styles in LaTeX. This key uses the control sequence parsed as its value to format the vector. For example, with the esvect package loaded with the style a:

```
\integral[
  double=S,
  variable=S,
  diff-vec=true,
  diff-vec-style=\vv,
  ]{\vv{S}}
```

$$\iint_S \vec{S} \cdot \mathrm{d}\vec{S} \tag{19}$$

4   Auxiliary macros

4.1   Loading new symbols

\NewIntegralSymbol {⟨*csname*⟩}

If you use a package which defines symbols that are not part of unicode-math, you can load them by using the \NewIntegralSymbol macro in the preamble. They will then be accessible via the int-symb key. The argument must take the forme of a macro *without backslash*: for example, \NewIntegralSymbol{myint}. Please note that \NewIntegralSymbol is only a means for the package to recognise pre-existing symbols and their corresponding macros internally. It is not a way to create a new glyph. It is the users responsibility to load the necessary packages so that the desired symbols are defined.

4.2 Looking back at limits

`{⟨keyword⟩}{⟨limits⟩}`

It is common to have to specify the same limits in an integral. To make them easier to write, the `limits(*)` key(s) also accept a keyword designating llimits defined 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` creates a keyword only if it doesn't already exist. No error message will be issued if it is the case.

- `\DeclareLimitsKeyword` creates a new keyword no matter what, overwritting any previous definitions.

Using one of the previous examples, we could modify the expression of the integral as follows:

```
\integral[
  limits={radius; height; circle},
  variable={\rho, z, \phi}
]{\rho z \phi}
```

$$\int_0^R \int_0^H \int_0^{2\pi} \rho z \phi \, \mathrm{d}\rho \, \mathrm{d}z \, \mathrm{d}\phi \tag{20}$$

Keyword	Limits
ab	a, b
real	$-\infty, +\infty$
positive	$0, +\infty$
negative	$-\infty, 0$
unit	$0, 1$
circle	$0, 2\pi$
scircle	$0, \pi$
qcircle	$0, \frac{pi}{2}$
height	$0, H$
radius	$0, R$
length	$0, L$
time	$0, T$

Table 1: List of keywords for limits.

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

4.3 Looking back at differentials

`\NewDifferentialKeyword`
`\RenewDifferentialKeyword`
`\ProvideDifferentialKeyword`
`\DeclareDifferentialKeyword`

`{⟨keyword⟩}{⟨differentials⟩}[⟨jacobian⟩]`

Similarly, the same groups of differentials often resurface. It is therefore also possible to define keywords whose comma list is predefined (the variants have the same behaviour). The sole difference is that we can also specify the Jacobian. The latter must also be a comma list, as it allows the package to split it when the `int-split` mode is enabled. Its display is controlled by the `jacobian` key as explained above.

```
\integral[
  triple={V},
  variable=cartesian
  ]{f(x, y, z)}
```

$$\iiint_V f(x, y, z)\, \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z \tag{21}$$

Keyword	Differentials	Jacobien
cartesian	x, y, z	–
polar	r, θ	r
cylindrical	r, θ, z	r
spherical	r, θ, ϕ	$r, \sin\theta$

Table 2: List of keywords for differentials.

4.4 Custom differentials placement

`\differentials` Although the option `invert-differentials` exists, it is often desirable to be able to place differentials wherever you want. For instance, they are often seen in the numerator of a fraction when the latter is equal to 1. To meet this need, the package provides the `\differentials` macro. When `int-split=false`, `\differentials` typesets all differentials in the same place. When `int-split=true`, `\differentials` must be rewritten between each semicolon. In both cases, the Jacobian will be attached to the differentials.

```
\integral[
  variable={a},
  ]{\frac{\differentials}{a}} = \ln(a) + C
```

$$\int \frac{\mathrm{d}a}{a} = \ln(a) + C \tag{22}$$