Introduction

At top level, Maple implements iterated integration by composing the \texttt{Int} or \texttt{int} commands. The older \texttt{student} package provided the \texttt{Doubleint} and \texttt{Tripleint} commands for iterated double and triple integrals, respectively. The \texttt{MultivariateCalculus} subpackage of the newer \texttt{Student} package provides a \texttt{MultiInt} command for writing iterated integrals, while the \texttt{VectorCalculus} package modifies the \texttt{int} command to accomplish the same thing. In addition, the \texttt{VectorCalculus} package contains the \texttt{SurfaceInt} and \texttt{Flux} commands, both capable of writing specialized iterated integrals.

In this article, we will contrast the syntax and applicability of the various commands Maple has for writing and evaluating iterated integrals.

Top Level

At top level, the inert iterated integral

\[
\int_{x_1}^{x_2} \int_{y_1(x)}^{y_2(x)} f(x,y) \, dy \, dx
\]

is formed with the syntax

\begin{verbatim}
> Int(Int(f(x,y), y=y[1](x)..y[2](x)), x=x[1]..x[2]);
\end{verbatim}
and evaluated with the `value` command. For example, to integrate

```maple
> f := x*y;
f := x*y
```

over $R$, the first-quadrant region bounded by the curves

```maple
> y1 := x^2;
y1 := x^2
> y2 := x;
y2 := x
```

we could write

```maple
> V := Int(Int(f, y=y1..y2), x=0..1);
V := \int_{0}^{1} \int_{x^2}^{x} \, xy \, dy \, dx
```

and evaluate the integral with

```maple
> value(V);
1
24
```

\textbf{Doubleint and Tripleint from the student Package}

Many long-time users of Maple are probably very familiar with the Doubleint and Tripleint commands from the student package. Despite their names, these commands implemented \textit{iterated} integrals, not \textit{multiple} integrals. (Multiple integrals exist that cannot be evaluated by iteration, and likewise, existence of the iterated integrals does not guarantee the multiple integral exists.)

Thus, the Doubleint command could be used to write the iterated integral

```maple
> student[Doubleint](f, y=y1..y2, x=0..1);
\int_{0}^{1} \int_{x^2}^{x} \, xy \, dy \, dx
```

Note that the ordering of the variables in Doubleint and Tripleint is consistent with that used when composing the Int command. It is the order in which one thinks out the bounds for the iterated integrals. The inner integral must have its bounds determined first, and that is the integration range that is supplied to either command first. The second set of bounds that must be thought out belong to the outer integral, and that range is entered second. I always considered this the "natural" order for writing the syntax of an iterated integral.

That the outer integral is to the left of the inner integral, as with all operator notation, makes for an inherent reversal of the ranges, just as the operator notation for partial derivatives reverses the order of the variables used in subscript notation. Thus, just as students must understand the notation
they likewise must understand that the order in which the ranges in an iterated integral are thought out is the opposite of the order in which they appear. However, the inner integral is both thought out and executed first, with the outer integral second in both regards.

Finally, recall that the DoubleInt and TripleInt commands always wrote the inert form of the iterated integral. Compared to the top-level usage, these commands not only saved writing a parenthesis or two, but also could be used for writing notation of the form

\[
\begin{align*}
\int_{x_1}^{x_2} \int_{y_1(x)}^{y_2(x)} \int_{z_1(x,y)}^{z_2(x,y)} f(x,y,z) \, dz \, dy \, dx
\end{align*}
\]

The MultiInt Command

An iterated integral can be written with the MultiInt command from the MultivariateCalculus subpackage of the Student package. This subpackage is made accessible with

\[
\begin{align*}
&> \text{restart;} \\
&> \text{with(Student[MultiInt])};
\end{align*}
\]

To write the iterated integral

\[
\int_{x=0}^{1} \int_{y_1}^{y_2} \int_{z_1}^{z_2} f(x,y,z) \, dz \, dy \, dx
\]

using the MultiInt command, the user must know that the order in which the ranges of integration are entered in the "natural" order described above. Thus, to integrate

\[
> f := x*y*z;
\]

over \( V \), the cylinder whose projection in the \( xy \)-plane is the region bounded by

\[
> y_1 := x^2;
\]

\[
> y_2 := x;
\]

and which is bounded below and above by the surfaces

\[
> z_1 := -4-x-y;
\]

\[
> z_2 := 4-x^2-y^2;
\]

use the syntax

\[
> \text{MultiInt}(f, z=z_1..z_2, y=y_1..y_2, x=0..1, output=integral);
\]
Notice that to obtain the integral, the special output parameter "integral" must be included. Without this parameter, the integration is immediate, as if the integral had been written with a composition of \texttt{int} commands.

\begin{verbatim}
> MultiInt(f, z=z1..z2, y=y1..y2, x=0..1);
\end{verbatim}

\begin{verbatim}
Writing notation such as
\[
\int_R \int g(x, y) \, dy \, dx
\]
\end{verbatim}

requires the more complex syntax

\begin{verbatim}
> MultiInt(g(x,y), y=`..R, x=`..`);
\end{verbatim}

where the space between the back quotes in the second range is essential. However, this inconvenience is counterbalanced by an increased functionality for handling polar, cylindrical, and spherical coordinates.

Thus, to integrate

\begin{verbatim}
> f := r*cos(3*theta)^2;
\end{verbatim}

\begin{verbatim}
f := r \cos(3 \theta)^2
\end{verbatim}

over the unit circle centered at the origin, one need only write

\begin{verbatim}
> MultiInt(f, r=0..1, theta=0..2*Pi, coordinates=polar[r,theta])
\end{verbatim}

\begin{verbatim}
\[ \frac{1}{3} \pi \]
\end{verbatim}

The inert form of this integral appears as

\begin{verbatim}
> q := MultiInt(f, r=0..1, theta=0..2*Pi, coordinates=polar[r,theta], output=integral);
\end{verbatim}

\begin{verbatim}
\[ q := \int_0^1 \int_0^{2\pi} r^2 \cos(3 \theta)^2 \, dr \, d\theta \]
\end{verbatim}

from which we see that Maple correctly maps the element of area \( dA = dy \, dx = dx \, dy \) to \( r \, dr \, d\theta \).

\section*{The Modified \texttt{int} Command in the \texttt{VectorCalculus} Package}

The \texttt{VectorCalculus} package modifies the top-level \texttt{int} command to give it the ability to write and evaluate iterated integrals of scalars, and single integrals of vectors. To see just how \texttt{int} performs in
this package, execute

> restart;

    interface(warnlevel=0):
    with(VectorCalculus):
    BasisFormat(false):

to access the package. (The BasisFormat command causes all vectors to be written as column vectors.)

Then, an iterated integral such as

\[
\int_{x_1}^{x_2} \int_{y_1}^{y_2(x)} f(x, y) \, dy \, dx
\]

or

\[
\int_{x_1}^{x_2} \int_{y_1(x)}^{y_2(x)} \int_{z_1(x,y)}^{z_2(x,y)} f(x, y, z) \, dz \, dy \, dx
\]

is implemented with the syntax

> int(f(x,y), \([x,y] = \text{Region}(x1..x2, y1(x)..y2(x)), \text{inert});

or

> int(f(x,y,z), \([x,y,z] = \text{Region}(x1..x2, y1(x)..y2(x), z1(x,y)..z2(x,y)), \text{inert});

Note that it is not possible to use subscripted variables in this instance because, for example, \(x_1\) is seen as the variable \(x\). Note also that the inert form of the integral is generated by using either the parameter "inert" or "output=integral." (The MultiInt command does not accept "inert"). Finally, note that the order of entry for the bounds on the integrals corresponds to operator notation, and is therefore the opposite of the "natural" order described above.

In addition to being able to integrate over general regions with a single call to \textbf{int}, it is possible to integrate over special pre-defined regions such as circles, spheres, triangles, rectangles, and ellipses. The following is a useful summary of such functionality.

\[
\text{int}(f, [x,y] = \text{Circle}(<a,b>,r))
\]

\[
\text{int}(f, [x,y] = \text{Ellipse}(x^2/a^2 + y^2/b^2 -1))
\]

\[
\text{int}(f, [x,y] = \text{Rectangle}(a..b, c..d))
\]

\[
\text{int}(f, [x,y] = \text{Triangle}(<a,b>, <c,d>, <p,q>))
\]

\[
\text{int}(f, [x,y] = \text{Sector}(\text{Circle}(<a,b>,r), \text{theta}[1], \text{theta}[2])
\]

\[
\text{int}(f, [x,y] = \text{Sector}(\text{Ellipse}(x^2/a^2 + y^2/b^2 - 1), \text{theta}[1], \text{theta}[2])
\]

\[
\text{int}(f, [x,y,z] = \text{Sphere}(<a,b,c>, r))
\]
To illustrate integration of
\[ f := x \cdot y; \]
over the triangle whose vertices are \((0, 0), (3, 5),\) and \((2, 4),\) use
\[ \int (f, [x,y] = \text{Triangle}(<0,0>, <3,5>, <2,4>)); \]
\[ \frac{17}{3} \]
To obtain this result without the special properties of the modified \texttt{int} command, begin by finding the equations of the lines between the vertices. These equations are
\[ \begin{align*}
Y_{12} & := \text{rhs(Student[Precalculus][Line]([0,0],[3,5])[1])}; \\
Y_{23} & := \text{rhs(Student[Precalculus][Line]([3,5],[2,4])[1])}; \\
Y_{31} & := \text{rhs(Student[Precalculus][Line]([2,4],[0,0])[1])};
\end{align*} \]
\[ \begin{align*}
Y_{12} & := \frac{5}{3} x \\
Y_{23} & := 2 + x \\
Y_{31} & := 2 x
\end{align*} \]
A graph of the region of integration is contained in Figure 1.
\[ \text{plot}([[x,Y_{12},x=0..3],[x,Y_{23},x=2..3],[x,Y_{31},x=0..2]], \text{color}=[\text{black,red,green}], \text{scaling=constrained, title="Figure 1"}); \]
The integration over the triangle must be written as the sum of the two iterated integrals
> q := Int(Int(f, y=Y12..Y31), x=0..2) + Int(Int(f, y=Y12..Y23), x=2..3);

\[
q := \int_0^2 \int_{\frac{5}{3}x}^{2x} xy \, dy \, dx + \int_2^3 \int_{\frac{5}{3}x}^{2 + x} xy \, dy \, dx
\]

whose value is
> value(q);

\[
\frac{17}{3}
\]

As a second example, consider the integral of
> f;

\[ xy \]

over the circle with center at (3, 4) and with radius 2, an iterated integral whose value is
> int(f, [x,y] = Circle(<3,4>, 2));

\[
48 \pi
\]
The inert form of this integral is returned as

\[ q := \int_{0}^{2\pi} \int_{0}^{2} x \left( x^2 \cos(y) \sin(y) + 4 x \cos(y) + 3 x \sin(y) + 12 \right) \, dy \, dx \]

where polar coordinates have been used. This is more easily seen if, instead of \( x \) and \( y \), we cause the integral to be written with the variables \( r \) and \( \theta \) by the brute-force substitution

\[ Q := \int_{0}^{2\pi} \int_{0}^{2} r \left( r^2 \cos(\theta) \sin(\theta) + 4 r \cos(\theta) + 3 r \sin(\theta) + 12 \right) \, d\theta \, dr \]

Of course, the integral still evaluates to

\[ \text{value}(Q) = 48\pi \]

## The `SurfaceInt` Command in the `VectorCalculus` Package

The surface integral of a scalar function is implemented with the `SurfaceInt` command in the `VectorCalculus` package. The surface integral of \( g(x, y, z) \) over that portion of the surface \( z = z(x, y) \) inside the cylinder with footprint \( R \), the region bounded by the curves \( y = y_1(x), y = y_2(x), x = a, \) and \( x = b \), is given by

\[ \text{SurfaceInt}(g(x,y,z), [x,y,z]=\text{Surface}(<x,y,z(x,y)>, x=a..b, y=y_1(x)..y_2(x)), \text{inert}) \]

Recognizing the surface-area element as

\[ d\sigma = \sqrt{1 + z_x^2 + z_y^2} \, dy \, dx \]

we see that this integral is just the "volume" under the surface \( g \sqrt{1 + z_x^2 + z_y^2} \) but inside the cylinder whose footprint in the \( xy \)-plane is the region \( R \). For example, if we take \( g(x, y) = 1 \) and the surface \( z = 1 \) with \( R \) bounded by the curves \( y_1 = x^2 \) and \( y_2 = x \), the surface integral is just the area of \( R \), given by the `SurfaceInt` command as

\[ \text{SurfaceInt}(1, [x,y,z]=\text{Surface}(<x,y,1>, x=0..1, y=x^2..x)); \]

\[ \frac{1}{6} \]

However, the area of \( R \) is also given by

\[ \text{Int}(x-x^2, x=0..1) = \int_{0}^{1} (x-x^2) \, dx = \frac{1}{6} \]
The SurfaceInt command also computes the surface integral of \( g(x, y, z) \) when the surface \( z = z(x, y) \) lies inside the cylinder whose cross-section \( R \) is a valid region recognized by the modified int command of the VectorCalculus package. For example, if to compute the surface integral of

\[
> f := x\cdot y\cdot z;
\]

on that portion of the surface

\[
> Z := 16 - x^2 - y^2;
\]

lying inside the cylinder whose cross-section is a circle with center \((2, 3)\) and radius 1, use

\[
> q := \text{evalf(SurfaceInt}(f, [x,y,z]=\text{Surface}(\langle x,y,Z\rangle, [x,y]=\text{Circle}(<2,3>,1)), \text{inert}));
\]

\[
q := 151.3361943
\]

The surface integral of \( g(x, y, z) \) over a surface described parametrically by

\[
x=x(u, v) \quad y=y(u, v)
\]

\[
z=z(u, v)
\]

\[
a \leq u \leq b
\]

\[
c \leq v \leq d
\]

is given by

\[
> \text{SurfaceInt}(g(x,y,z), [x,y,z]=\text{Surface}(\langle x(u,v), y(u,v), z(u,v)\rangle, u=a..b, v=c..d), \text{inert});
\]

\[
\int_{a}^{b} \int_{c}^{d} g(x(u,v), y(u,v), z(u,v))
\]

\[
\left(\left(\frac{\partial}{\partial u} y(u,v)\right) \left(\frac{\partial}{\partial v} z(u,v)\right) - \left(\frac{\partial}{\partial u} z(u,v)\right) \left(\frac{\partial}{\partial v} y(u,v)\right)\right)^2
\]

\[
+ \left(\left(\frac{\partial}{\partial u} z(u,v)\right) \left(\frac{\partial}{\partial v} x(u,v)\right) - \left(\frac{\partial}{\partial u} x(u,v)\right) \left(\frac{\partial}{\partial v} z(u,v)\right)\right)^2
\]

\[
+ \left(\left(\frac{\partial}{\partial u} x(u,v)\right) \left(\frac{\partial}{\partial v} y(u,v)\right) - \left(\frac{\partial}{\partial u} y(u,v)\right) \left(\frac{\partial}{\partial v} x(u,v)\right)\right)^2\right)^{1/2}
\]

\[
d\sigma = \sqrt{J_1^2 + J_2^2 + J_3^2}
\]

where \( J_1, J_2, J_3 \) are, respectively, the Jacobians

\[
J_1 = \det \text{Matrix}([\ [y, y], [z, z]] = y_u z_v - y_v z_u
\]

\[
J_2 = \det \text{Matrix}([\ [z, z], [x, x]] = z_u x_v - z_v x_u
\]
\[ J_3 = \det \begin{bmatrix} [x_u, x_v] \end{bmatrix} \begin{bmatrix} [y_u, y_v] \end{bmatrix} = x_u y_v - x_v y_u \]

which can be obtained in Maple via the syntax

```maple
> Jacobian([y(u,v),z(u,v)],[u,v], determinant);
Jacobian([z(u,v),x(u,v)],[u,v], determinant);
Jacobian([x(u,v),y(u,v)],[u,v], determinant);
```

\[
\begin{bmatrix}
\frac{\partial}{\partial u} y(u, v) & \frac{\partial}{\partial v} y(u, v) \\
\frac{\partial}{\partial u} z(u, v) & \frac{\partial}{\partial v} z(u, v) \\
\frac{\partial}{\partial u} x(u, v) & \frac{\partial}{\partial v} x(u, v)
\end{bmatrix}
\left( \frac{\partial}{\partial u} y(u, v) \right) \left( \frac{\partial}{\partial v} z(u, v) \right) - \left( \frac{\partial}{\partial u} z(u, v) \right) \left( \frac{\partial}{\partial v} y(u, v) \right)
```

If a sphere of radius \( a > 0 \) is described in spherical coordinates by \( 0 \leq \phi \leq \pi, 0 \leq \theta \leq 2\pi \), then its surface area is given by

```maple
> SurfaceInt(1, [x,y,z] = Surface(<a,phi,theta>, phi=0..Pi, theta=0..2*Pi, coords=spherical)) assuming a>0;
4 \pi a^2
```

Alternatively, the surface area of a sphere of radius \( a \) could also be found with

```maple
> SurfaceInt(1, [x,y,z] = Sphere(<0,0,0>, a ));
4 \pi a^2
```

In the first calculation, the parametrization was in terms of spherical coordinates, while in the second, Maple understood one of two pre-defined surfaces. The second such surface is the box.

To obtain the surface integral of the scalar \( g(x, y, z) \) over the surface of a rectangular box, the syntax

```maple
> SurfaceInt(g(x,y,z), [x,y,z] = Box(a..b, c..d, r..s), inert);
```

\[
\int_{r=a}^{r=b} \int_{c}^{d} \int_{t}^{s} (g(a, s, t) + g(b, s, t)) \, ds \, dt + \int_{r}^{r=a} \int_{c}^{d} \int_{t}^{s} (g(s, c, t) + g(s, d, t)) \, ds \, dt + \int_{r}^{r=a} \int_{c}^{d} \int_{t}^{s} (g(s, t, r) + g(s, t, s)) \, ds \, dt
\]

can be used. Each of the three integrals returned by Maple combines the contributions from opposite faces of the box.

\section*{The \textit{Flux} Command in the \textit{VectorCalculus} Package}

The \textit{Flux} command in the \textit{VectorCalculus} package will compute the flux (surface integral of the
normal component of the field) of a vector field through a surface. If the surface is closed, the user has the choice of implementing the flux integral with either the inward or outward normal. Thus, the flux of the vector field

\[
F := \text{VectorField}(\langle x, y, z \rangle / \sqrt{x^2+y^2+z^2}, \text{cartesian}[x,y,z]);
\]

through the surface of the unit sphere centered at the origin can be computed as either of

\[
\text{Flux}(F, \text{Sphere}(\langle 0,0,0 \rangle, 1))
\]

\[
\text{Flux}(F, \text{Sphere}(\langle 0,0,0 \rangle, 1, \text{inward}))
\]

In the first case, the outward normal is used, but in the second, the inward normal is used.

No such control over the normal is provided for an open surface. Maple arbitrarily picks a normal direction and provides neither for changing its orientation, nor for discovering which orientation was used. Thus, to integrate the vector field \( F \) over the surface of the upper hemisphere, use

\[
\text{Flux}(F, \text{Surface}(\langle a,\phi,\theta \rangle, \phi=0..\pi/2, \theta=0..2\pi, \text{coords=spherical})) \text{ assuming } a>0;
\]

\[
2 \pi a^2
\]

From the value returned, it is obvious that the upward normal (outward on the closed sphere) was used. But in general, it would be difficult to tell which way the net flux was directed for an arbitrary field and (open) surface.

A calculation of the flux of \( F \) through the upper hemisphere can be obtained from first principles in which the direction of the normal would be known. From the value of the flux obtained this way, it would be possible to infer the direction of the normal field used by Maple's \texttt{Flux} command.

To this end, describe the upper hemisphere explicitly with

\[
Z := \sqrt{a^2 - x^2 - y^2};
\]

so that a normal field on the surface is given by

\[
\text{N1 := } \langle -\text{diff}(Z,x), -\text{diff}(Z,y), 1 \rangle;
\]
To obtain a unit normal field, the `Normalize` command from the `LinearAlgebra` package is needed. Hence, execute

\[ \text{with(LinearAlgebra):} \]

and then

\[ \text{N2 := Normalize(N1,2);} \]

\[ N2 := \begin{bmatrix} \frac{x}{\sqrt{a^2-x^2-y^2}} \\
\sqrt{1 + \left(\frac{x}{\sqrt{a^2-x^2-y^2}}\right)^2 + \left(\frac{y}{\sqrt{a^2-x^2-y^2}}\right)^2} \sqrt{a^2-x^2-y^2} \\
\frac{y}{\sqrt{a^2-x^2-y^2}} \\
\sqrt{1 + \left(\frac{x}{\sqrt{a^2-x^2-y^2}}\right)^2 + \left(\frac{y}{\sqrt{a^2-x^2-y^2}}\right)^2} \sqrt{a^2-x^2-y^2} \\
\frac{1}{\sqrt{1 + \left(\frac{x}{\sqrt{a^2-x^2-y^2}}\right)^2 + \left(\frac{y}{\sqrt{a^2-x^2-y^2}}\right)^2}} \end{bmatrix} \]

Of course, this calculation is valid in Cartesian coordinates only.

Further simplification of the unit normal field is obtained with

\[ \text{N := simplify(N2, symbolic);} \]

\[ N := \begin{bmatrix} \frac{x}{a} \\
\frac{y}{a} \\
\sqrt{a^2-x^2-y^2} \\
\frac{a}{\sqrt{1 + \left(\frac{x}{\sqrt{a^2-x^2-y^2}}\right)^2 + \left(\frac{y}{\sqrt{a^2-x^2-y^2}}\right)^2}} \end{bmatrix} \]

from which it's clear that the normal is upward (outward on the closed sphere).

To obtain \( F \cdot N \), use

\[ \text{FN := simplify(DotProduct(F,N, conjugate=false));} \]

\[ FN := \frac{x^2+y^2+z\sqrt{a^2-x^2-y^2}}{\sqrt{x^2+y^2+z^2} \cdot a} \]
The flux of $\mathbf{F}$ through the upper hemisphere is the surface integral of $\mathbf{F} \cdot \mathbf{N}$, obtained in Maple with

```maple
> q := SurfaceInt(F*N, [x,y,z]=Surface(<x,y,Z>, x=-a..a, y=-sqrt(a^2-x^2)..sqrt(a^2-x^2)));
```

To simplify the integrand of this integral, extract is via

```maple
> f := op([1,1],q);
f:= a
```

and simplify it via

```maple
> f1 := simplify(f) assuming a>0
f1 := a
```

The surface integral is merely the integral of this quantity over a disk with center at the origin, with radius $a$, and lying in the $xy$-plane. This integral can be obtained in Maple with

```maple
> int(f1, [x,y]=Circle(<0,0>,a)) assuming a>0;
```

which agrees with the result of the $\text{Flux}$ command. Hence, the $\text{Flux}$ command used the upward (outward on the closed sphere) normal, as we did in our calculation from first principles.

Legal Notice: The copyright for this application is owned by Maplesoft. The application is intended to demonstrate the use of Maple to solve a particular problem. It has been made available for product evaluation purposes only and may not be used in any other context without the express permission of Maplesoft.