Maxima by Example: Ch.7: Symbolic Integration *

Edwin L. Woollett

September 16, 2010

Contents

*This version uses **Maxima 5.18.1**. This is a live document. Check `http://www.csulb.edu/~woollett/` for the latest version of these notes. Send comments and suggestions to `woollett@charter.net`

Feedback from readers is the best way for this series of notes to become more helpful to new users of Maxima. *All* comments and suggestions for improvements will be appreciated and carefully considered.

```
The Maxima session transcripts were generated using the XMaxima
graphics interface on a Windows XP computer, and copied into
a fancy verbatim environment in a latex file which uses the
fancyvrb and color packages.

We use qdraw.mac (available on the author's web page with Ch. 5 materials)
for plots.
```


```
Maxima.sourceforge.net. Maxima, a Computer Algebra System. Version 5.18.1
 (2009). http://maxima.sourceforge.net/
```

7.1 Symbolic Integration with integrate

Although the process of finding an integral can be viewed as the inverse of the process of finding a derivative, in practice finding an integral is more difficult. It turns out that the integral of fairly simple looking functions cannot be expressed in terms of well known functions, and this has been one motivation for the introduction of definitions for a variety of ”special functions”, as well as efficient methods for numerical integration (quadrature) discussed in Ch. 8.

In the following, we always assume we are dealing with a real function of a real variable and that the function is single-valued and continuous over the intervals of interest.

The Maxima manual entry for **integrate** includes (we have made some changes and additions):

> Function: **integrate(expr, var)**
> Function: **integrate(expr, var, a, b)**
> Attempts to symbolically compute the integral of **expr** with respect to **var**. **integrate(expr, var)** is an indefinite integral, while **integrate(expr, var, a, b)** is a definite integral, with limits of integration **a** and **b**. The integral is returned if **integrate** succeeds. Otherwise the return value is the noun form of the integral (the quoted operator **’integrate**) or an expression containing one or more noun forms. The noun form of **integrate** is displayed with an integral sign if **display2d** is set to **true** (which is the default).
>
> In some circumstances it is useful to construct a noun form by hand, by quoting **integrate** with a single quote, e.g., **’integrate(expr, var)**. For example, the integral may depend on some parameters which are not yet computed. The noun may be applied to its arguments by **ev(iexp, nouns)** where **iexp** is the noun form of interest.
>
> The Maxima function **integrate** is defined by the lisp function **$integrate** in the file /src/simp.lisp. The indefinite integral invocation, **integrate(expr,var)**, results in a call to the lisp function **sinint**, defined in src/sin.lisp, unless the flag **risch** is present, in which case the lisp function **rischint**, defined in src/risch.lisp, is called. The definite integral invocation, **integrate(expr,var,a,b)**, causes a call to the lisp function **$defint**, defined in src/defint.lisp. The lisp function **$defint** is available as the Maxima function **defint** and can be used to bypass **integrate** for a definite integral.
>
> To integrate a Maxima function $\mathbf{f(x)}$, insert $\mathbf{f(x)}$ in the **expr** slot.
>
> **integrate** does not respect implicit dependencies established by the **depends** function.
>
> **integrate** may need to know some property of the parameters in the integrand. **integrate** will first consult the **assume** database, and, if the variable of interest is not there, **integrate** will ask the user. Depending on the question, suitable responses are **yes;** or **no;**, or **pos;**, **zero;**, or **neg;**. Thus, the user can use the **assume** function to avoid all or some questions.

7.2 Integration Examples and also defint, ldefint, beta, gamma, erf, and logabs

Example 1

Our first example is the indefinite integral $\int \sin^3 x \, dx$:

```
(%i1) integrate (sin(x)^3, x);
                                    3
                                 cos  (x)
(%o1)                            ------- - cos(x)
                                    3
(%i2) ( diff (%,x), trigsimp(%%) );
                                         3
(%o2)                                 sin  (x)
```

Notice that the indefinite integral returned by **integrate** *does not include* the arbitrary constant of integration which can always be added.

If the returned integral is correct (up to an arbitrary constant), then the first derivative of the returned indefinite integral should be the original integrand, although we may have to simplify the result manually (as we had to do above).

Example 2

Our second example is another indefinite integral, $\int x \, (b^2 - x^2)^{-1/2} \, dx$:

```
(%i3) integrate (x/ sqrt (b^2 - x^2), x);
                                            2    2
(%o3)                             - sqrt(b   - x )
(%i4) diff(%,x);
                                         x
(%o4)                              -------------
                                         2    2
                                   sqrt(b   - x )
```

Example 3

The definite integral can be related to the "area under a curve" and is the more accessible concept, while the integral is simply a function whose first derivative is the original integrand.

Here is a definite integral, $\int_0^\pi \cos^2 x \, e^x \, dx$:

```
(%i5) i3 : integrate (cos(x)^2 * exp(x), x, 0, %pi);

                                       %pi
                                    3 %e       3
(%o5)                              ------- - -
                                       5       5
```

Instead of using **integrate** for a definite integral, you can try **ldefint** (think Limit definite integral), which may provide an alternative form of the answer (if successful).

From the Maxima manual:

Function: **ldefint(expr, x, a, b)**
Attempts to compute the definite integral of **expr** by using **limit** to evaluate the indefinite integral of **expr** with respect to x at the upper limit b and at the lower limit a. If it fails to compute the definite integral, **ldefint** returns an expression containing limits as noun forms.

ldefint is not called from **integrate**, so executing **ldefint(expr, x, a, b)** may yield a different result than **integrate(expr, x, a, b)**. **ldefint** always uses the same method to evaluate the definite integral, while **integrate** may employ various heuristics and may recognize some special cases.

Here is an example of use of **ldefint**, as well as the direct use of **defint** (which bypasses **integrate**):

```
(%i6) ldefint (cos(x)^2 * exp(x), x, 0, %pi);
                                         %pi
                                     3 %e       3
(%o6)                                ------- - -
                                        5       5
(%i7) defint (cos(x)^2 * exp(x), x, 0, %pi);
                                         %pi
                                     3 %e       3
(%o7)                                ------- - -
                                        5       5
```

Example 4

Here is an example of a definite integral over an infinite range, $\int_{-\infty}^{\infty} x^2\, e^{-x^2}\, dx$:

```
(%i8) integrate (x^2 * exp(-x^2), x, minf, inf);
                                      sqrt(%pi)
(%o8)                                 ---------
                                          2
```

To check this integral, we first ask for the indefinite integral and then check it by differentiation.

```
(%i9) i1 : integrate(x^2*exp(-x^2),x );
                                                         2
                                                      - x
                           sqrt(%pi) erf(x)     x %e
(%o9)                     ---------------- - --------
                                  4                2
(%i10) diff(i1,x);
                                                2
                                       2    - x
(%o10)                                 x   %e
```

Thus the indefinite integral is correct. The second term, heavily damped by the factor e^{-x^2} at $\pm\infty$, does not contribute to the definite integral. The first term is proportional to the (Gauss) error function, $\mathrm{erf}(x)$, in which x is real. For (in general) complex $w = u + i\, v$,

$$\mathrm{Erf}(w) = \frac{2}{\sqrt{\pi}} \int_0^w e^{-z^2}\, dz \tag{7.1}$$

in which we can integrate over any path connecting 0 and w in the complex $z = x + i\, y$ plane, since the integrand is an entire function of z (no singularities in the finite z plane.

Let's make a plot of **erf(x)**:

```
(%i11) ( load(draw),load(qdraw) )$
(%i12) qdraw(yr(-2,2),ex1(erf(x),x,-5,5,lw(5),lc(red),lk("erf(x)") ) )$
```

with the result:

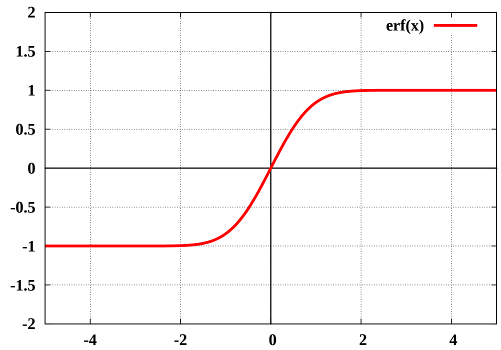

Figure 1: $\mathrm{erf}(x)$

Maxima's **limit** function confirms what the plot indicates:

```
(%i13) [limit(erf(x),x, inf), limit(erf(x),x, minf)];
(%o13)                          [1, - 1]
```

Using these limits in **%o9** produces the definite integral desired.

Example 5: Use of assume

We next calculate the definite integral $\int_0^\infty x^a \, (x+1)^{-5/2} \, dx$.

```
(%i1) (assume(a>1),facts());
(%o1)                          [a > 1]
(%i2) integrate (x^a/(x+1)^(5/2), x, 0, inf );
   2 a + 2
Is ------- an integer?
      5

no;
Is  2 a - 3  positive, negative, or zero?

neg;
                                3
(%o2)                     beta(- - a, a + 1)
                                2
```

The combination of $\mathrm{assume}(a > 1)$ and $2\,a - 3 < 0$ means that we are assuming $1 < a < 3/2$. These assumptions about a imply that $4/5 < (2\,a + 2)/5 < 1$. To be consistent, we must hence answer **no** to the first question.

Let's tell Maxima to **forget** about the **assume** assignment and see what the difference is.

```
(%i3) ( forget(a>1), facts() );
(%o3)                                  []
(%i4) is( a>1 );
(%o4)                                unknown
(%i5) integrate (x^a/(x+1)^(5/2), x, 0, inf );
Is  a + 1  positive, negative, or zero?

pos;
Is a an integer?

no;
      7
Is ------- an integer?
   2 a + 4

no;
Is  2 a - 3  positive, negative, or zero?

neg;
                                     3
(%o5)                           beta(- - a, a + 1)
                                     2
(%i6) [is( a>1 ), facts() ];
(%o6)                             [unknown, []]
```

Thus we see that omitting the initial `assume(a>1)` statement causes **integrate** to ask four questions instead of two. We also see that answering questions posed by the **integrate** dialogue script does not result in population of the **facts** list.

The Maxima **beta** function has the manual entry:

```
Function: beta (a, b)
The beta function is defined as gamma(a) gamma(b)/gamma(a+b) (A&S 6.2.1).
```

In the usual mathematics notation, the beta function can be defined in terms of the gamma function as

$$B(r, s) = \frac{\Gamma(r)\,\Gamma(s)}{\Gamma(r + s)} \tag{7.2}$$

for all r, s in the complex plane.

The Maxima **gamma** function has the manual entry

```
Function: gamma (z)
The basic definition of the gamma function (A&S 6.1.1) is

                                 inf
                                /
                                [     z - 1   - t
                   gamma(z) =   I    t       %e    dt
                                ]
                                /
                                 0
```

The gamma function can be defined for complex z and $Re(z) > 0$ by the integral along the real t axis

$$\Gamma(z) = \int_0^\infty t^{z-1}\, e^{-t}\, d\, t \tag{7.3}$$

and for $Im(z) = 0$ and $Re(z) = n$ and n an integer greater than zero we have

$$\Gamma(n + 1) = n\,! \tag{7.4}$$

How can we check the definite integral Maxima has offered? If we ask the **integrate** function for the indefinite integral, we get the "noun form", a signal of failure:

```
(%i7) integrate(x^a/(x+1)^(5/2), x );
                                /      a
                                [     x
(%o7)                          I ---------- dx
                                ]        5/2
                                / (x + 1)
(%i8) grind(%)$
'integrate(x^a/(x+1)^(5/2),x)$
```

Just for fun, let's include the **risch** flag and see what we get:

```
(%i9) integrate(x^a/(x+1)^(5/2), x ), risch;
                         /           a log(x)
                         [         %e
(%o9)                    I -------------------------- dx
                         ]                2
                         / sqrt(x + 1) (x  + 2 x + 1)
```

We again are presented with a noun form, but the integrand has been written in a different form, in which the identity

$$x^A = e^{A\ln(x)}$$

has been used.

We can at least make a spot check for a value of the parameter a in the middle of the assumed range $(1, 3/2)$, namely for $a = 5/4$.

```
(%i10) float(beta(1/4,9/4));
(%o10)                        3.090124462168955
(%i11) quad_qagi(x^(5/4)/(x+1)^(5/2),x, 0, inf);
(%o11)        [3.090124462010259, 8.6105700347616221E-10, 435, 0]
```

We have used the quadrature routine, **quad_qagi** (see Ch. 8) for a numerial estimate of this integral. The first element of the returned list is the numerical answer, which agrees with the analytic answer.

Example 6: Automatic Change of Variable and gradef

Here is an example which illustrates Maxima's ability to make a change of variable to enable the return of an indefinite integral. The task is to evaluate the indefinite integral

$$\int \frac{\sin(r^2)\, dr(x)/dx}{q(r)}\, dx \tag{7.5}$$

by telling Maxima that the $\sin(r^2)$ in the numerator is related to $q(r)$ in the denominator by the derivative:

$$\frac{d\, q(u)}{d\, u} = \sin(u^2). \tag{7.6}$$

We would manually rewrite the integrand (using the chain rule) as

$$\frac{\sin(r^2)\, dr(x)/dx}{q} = \frac{1}{q}\,(dq(r)/dr)\,(dr(x)/dx) = \frac{1}{q}\frac{dq}{dx} = \frac{d}{dx}\ln(q) \tag{7.7}$$

and hence obtain the indefinite integral $\ln(q(r(x)))$.

Here we assign the derivative knowledge using **gradef** (as discussed in Ch. 6):

```
(%i1) gradef(q(u),sin(u^2) )$
(%i2) integrand : sin(r(x)^2)* 'diff(r(x),x ) /q( r(x) ) ;
                              d                2
                             (-- (r(x))) sin(r  (x))
                              dx
(%o2)                        ----------------------
                                   q(r(x))
(%i3) integrate(integrand,x);
(%o3)                            log(q(r(x)))
(%i4) diff(%,x);
                              d                2
                             (-- (r(x))) sin(r  (x))
                              dx
(%o4)                        ----------------------
                                   q(r(x))
```

Note that **integrate** pays no attention to **depend** assignments, so the briefer type of notation which **depend** allows with differentiation cannot be used with **integrate**:

```
(%i5) depends(r,x,q,r);
(%o5)                             [r(x), q(r)]
(%i6) integrand : sin(r^2)* 'diff(r,x) / q;
                                    dr      2
                                    -- sin(r )
                                    dx
(%o6)                               ----------
                                        q
(%i7) integrate(integrand,x);
                                          /
                                       2  [ dr
                                 sin(r ) I -- dx
                                          ] dx
                                          /
(%o7)                            ---------------
                                        q
```

which is fatally flawed, since Maxima pulled both $\sin(r(x)^2)$ and $1/q(r(x))$ outside the integral.

Of course, the above **depends** assignment will still allow Maxima to rewrite the derivative of q with respect to x using the chain rule:

```
(%i8) diff(q,x);
                                     dq dr
(%o8)                                -- --
                                     dr dx
```

Example 7: Integration of Rational Algebraic Functions, rat, ratsimp, and partfrac

A *rational algebraic function* can be written as a quotient of two polynomials. Consider the following function of x.

```
(%i1) e1 : x^2 + 3*x -2/(3*x)+110/(3*(x-3)) + 12;
                           2           2       110
(%o1)                     x  + 3 x - --- + --------- + 12
                                     3 x   3 (x - 3)
```

We can obviously find the lowest common denominator and write this as the ratio of two polynomials, using either **rat** or **ratsimp**.

```
(%i2) e11 : ratsimp(e1);
                                    4      2
                                   x  + 3 x  + 2
(%o2)                              -------------
                                      2
                                     x  - 3 x
```

Because the polynomial in the numerator is of higher degree than the polynomial in the denominator, this is called an *improper rational fraction*. Any improper rational fraction can be reduced by division to a mixed form, consisting of a sum of some polynomial and a sum of proper fractions. We can recover the "partial fraction" representation in terms of *proper rational fractions* (numerator degree less than denominator degree) by using **partfrac**`(expr, var)`.

```
(%i3) e12 : partfrac(e11,x);
                           2           2       110
(%o3)                     x  + 3 x - --- + --------- + 12
                                     3 x   3 (x - 3)
```

With this function of x expressed in partial fraction form, you are able to write down the indefinite integral immediately (ie., by inspection, without using Maxima). But, of course, we want to practice using Maxima!

```
(%i4) integrate(e11,x);
                                                      3      2
                2 log(x)   110 log(x - 3)   2 x  + 9 x  + 72 x
(%o4)         - -------- + -------------- + ------------------
                   3              3                  6
(%i5) integrate(e12,x);
                                                     3      2
                 2 log(x)   110 log(x - 3)    x    3 x
(%o5)          - -------- + -------------- + -- + ---- + 12 x
                    3              3          3     2
```

Maxima has to do less work if you have already provided the partial fraction form as the integrand; otherwise, Maxima internally seeks a partial fraction form in order to do the integral.

Example 8

The next example shows that **integrate** can sometimes split the integrand into at least one piece which can be integrated, and leaves the remainder as a formal expression (using the noun form of **integrate**). This may be possible if the denominator of the integrand is a polynomial which Maxima can factor.

```
(%i6) e2: 1/(x^4 - 4*x^3 + 2*x^2 - 7*x - 4);
                                     1
(%o6)                    --------------------------
                           4      3      2
                          x  - 4 x  + 2 x  - 7 x - 4
(%i7) integrate(e2,x);
                                    /  2
                                    [ x  + 4 x + 18
                                    I ------------- dx
                                    ]  3
                        log(x - 4)  / x  + 2 x + 1
(%o7)                  ---------- - ------------------
                            73               73
(%i8) grind(%)$
log(x-4)/73-('integrate((x^2+4*x+18)/(x^3+2*x+1),x))/73$
(%i9) factor(e2);
                                     1
(%o9)                      ----------------------
                                     3
                           (x - 4) (x  + 2 x + 1)
(%i10) partfrac(e2,x);
                                       2
                            1         x  + 4 x + 18
(%o10)                  ---------- - ----------------
                        73 (x - 4)          3
                                     73 (x  + 2 x + 1)
```

We have first seen what Maxima can do with this integrand, using the **grind** function to clarify the resulting expression, and then we have used **factor** and **partfrac** to see how the split-up arises. Despite a theorem that the integral of every rational function can be expressed in terms of algebraic, logarithmic and inverse trigonometric expressions, Maxima declines to return a symbolic expression for the second, formal, piece of this integral (which is good because the exact symbolic answer is an extremely long expression).

Example 9: The logabs Parameter and log

There is a global parameter **logabs** whose default value is **false** and which affects what is returned with an indefinite integral containing logs.

```
(%i11) logabs;
(%o11)                              false
(%i12) integrate(1/x,x);
(%o12)                             log(x)
(%i13) diff(%,x);
                                     1
(%o13)                               -
                                     x
(%i14) logabs:true$
```

```
(%i15) integrate(1/x,x);
(%o15)                          log(abs(x))
(%i16) diff(%,x);
                                     1
(%o16)                               -
                                     x
(%i17) log(-1);
(%o17)                           log(- 1)
(%i18) float(%);
(%o18)                    3.141592653589793 %i
```

When we override the default and set **logabs** to **true**, the argument of the **log** function is wrapped with **abs**. According to the manual

> When doing indefinite integration where logs are generated, e.g. integrate(1/x,x), the answer is given in terms of log(abs(...)) if logabs is true, but in terms of log(...) if logabs is false. For definite integration, the logabs:true setting is used, because here ”evaluation” of the indefinite integral at the endpoints is often needed.

7.3 Piecewise Defined Functions and integrate

We can use Maxima's **if**, **elseif**, and **else** construct to define piecewise defined functions. We first define and plot a square wave of unit height which extends from $1 \le x \le 3$.

```
(%i1) u(x) := if x >= 1 and x <= 3 then 1 else 0$
(%i2) map('u,[0.5,1,2,3,3.5]);
(%o2)                          [0, 1, 1, 1, 0]
(%i3) (load(draw),load(qdraw))$
               qdraw(...), qdensity(...), syntax: type qdraw();
(%i4) qdraw( yr(-1,2),ex1(u(x),x,0,4,lw(5),lc(blue)) )$
```

This produces

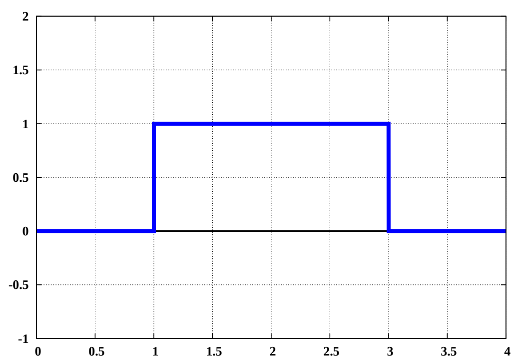

Figure 2: if $x > 1$ and $x < 3$ then 1 else 0

We next define a function which is $(x - 1)$ for $1 \le x < 2$ and is $(6 - 2x)$ for $2 \le x \le 3$ and is 0 otherwise.

```
(%i5) g(x):= if x >= 1 and x < 2 then (x-1)
              elseif x >= 2 and x <= 3 then (6 - 2*x) else 0$
(%i6) map('g,[1/2,1,3/2,2,5/2,3,7/2]);
                                    1
(%o6)                        [0, 0, -, 2, 1, 0, 0]
                                    2
(%i7) qdraw( yr(-1,3),ex1(g(x),x,0,4,lw(5),lc(blue)) )$
```

which produces

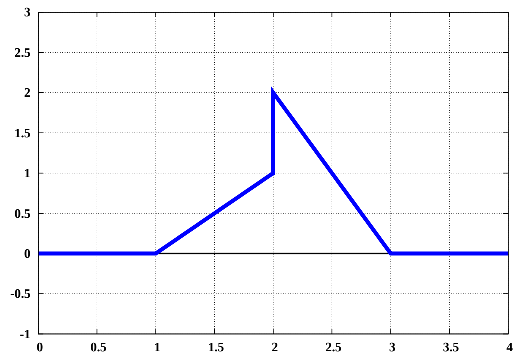

Figure 3: Example of a Piecewise Defined Function

This is not quite the figure we want, since it doesn't really show the kind of discontinuity we want to illustrate. With our definition of $g(x)$, **draw2d** draws a blue vertical line from $(2, 1)$ to $(2, 2)$. We can change the way $g(x)$ is plotted to get the plot we want, as in

```
(%i8) block([small :1.0e-6],
         qdraw( yr(-1,3),ex1(g(x),x,0,2-small,lw(5),lc(blue)),
                   ex1(g(x),x,2+small,4,lw(5),lc(blue)) ) )$
```

which produces

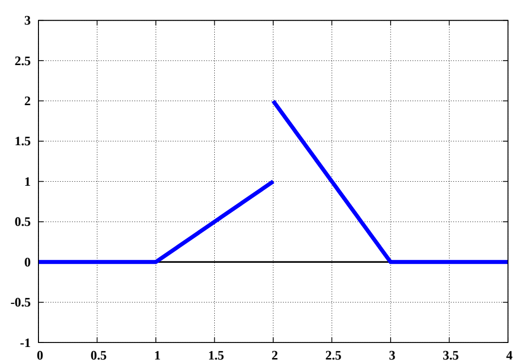

Figure 4: Example with More Care with Plot Limits

The Maxima function **integrate** cannot handle correctly a function defined with the **if**, **elseif**, and **else** constructs.

```
(%i9) integrate(g(x),x,0,4);
        4
       /
       [
(%o9) I   (if (x > 1) and (x < 2) then x - 1 elseif (x > 2) and (x < 3)
       ]
       /
        0
                                                       then 6 - 2 x else 0) dx
```

This means that for now we will have to break up the integration interval into sub-intervals by hand and add up the individual integral results.

```
(%i10) integrate(x-1,x,1,2) + integrate(6-2*x,x,2,3);
                                              3
(%o10)                                        -
                                              2
```

7.4 Area Between Curves Examples

Example 1

We will start with a very simple example, finding the area between $f_1(x) = \sqrt{x}$ and $f_2(x) = x^{3/2}$. A simple plot shows that the curves cross at $x = 0$ and $x = 1$.

```
(%i1) (load(draw),load(qdraw) )$
(%i2) f1(x) := sqrt(x)$
(%i3) f2(x) := x^(3/2)$
(%i4) qdraw(   xr(-.5,1.5), yr(-.5,1.5),
         ex1(f1(x),x,0,1.5,lw(5),lc(blue) ),
         ex1(f2(x),x,0,1.5,lw(5),lc(red) ) )$
```

This produces the plot:

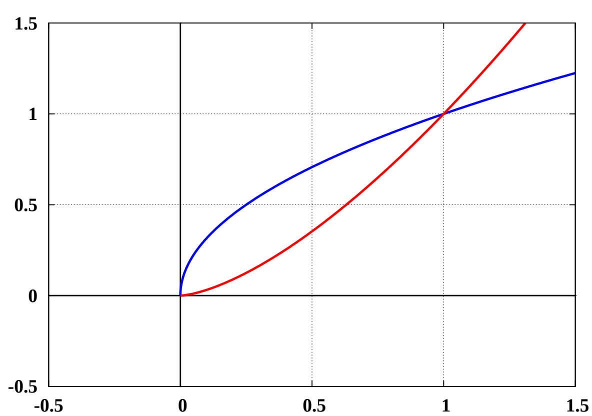

Figure 5: $\sqrt{x}$ (blue) and $x^{3/2}$ (red)

Next we redraw this simple plot, but add some shading in color to show the area. The simplest way to do this is to draw some vertical lines between the functions in the region of interest $0 \le x \le 1$. We can use the **qdraw** package function **line**. We first construct a list of x axis positions for the vertical lines, using values $0.01, 0.02, ...0.99$. We then construct a list `vv` of the vertical lines and merge that list with a list `qdlist` containing the rest of the plot instructions. We then use **apply** to pass this list as a set of arguments to **qdraw**.

```
(%i5) qdlist : [xr(-.5,1.5), yr(-.5,1.5),
         ex1(f1(x),x,0,1.5,lw(5),lc(blue) ) ,
         ex1(f2(x),x,0,1.5,lw(5),lc(red) ) ]$
(%i6) xv:float(makelist(i,i,1,99)/100)$
(%i7) (vv:[],for x in xv do
        vv:cons(line(x,f2(x),x,f1(x),lw(1),lc(khaki) ),vv),
        vv:reverse(vv) )$
(%i8) qdlist : append(vv,qdlist)$
(%i9) apply('qdraw, qdlist)$
```

which produces the result

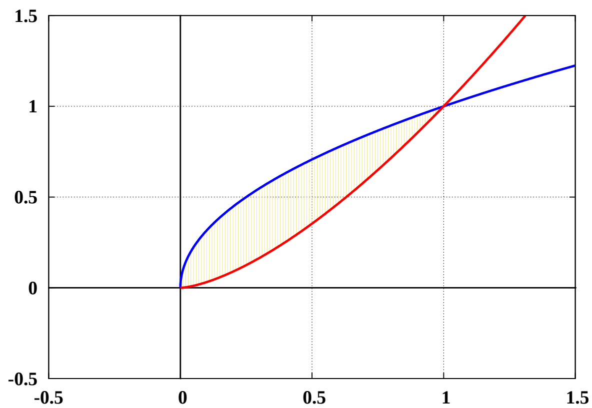

Figure 6: $\sqrt{x}$ (blue) and $x^{3/2}$ (red)

If we did not know the intersection location of the two curves, we could use **solve** or **`find_root`** for example.

```
(%i10) solve( f1(x) = f2(x),x );
(%o10)                          [x = 0, x = 1]
```

Once we know the interval to use for adding up the area and we know that in this interval $f_1(x) > f_2(x)$, we simply sum the infinitesimal areas given by $(f_1(x) - f_2(x))\,dx$ (base dx columns) over the interval $0 \le x \le 1$.

```
(%i11) integrate(f1(x) - f2(x),x,0,1);
                                           4
(%o11)                                    ----
                                           15
```

so the area equals $4/15$.

$$\int_0^1 (\sqrt{x} - x^{3/2})\,dx = 4/15 \tag{7.8}$$

Example 2

As a second example we consider two polynomial functions:
$f_1(x) = (3/10)\,x^5 - 3\,x^4 + 11\,x^3 - 18\,x^2 + 12\,x + 1$
and $f_2(x) = -4\,x^3 + 28\,x^2 - 56\,x + 32$. We first make a simple plot for orientation.

```
(%i1) f1(x) := (3/10)*x^5 -3*x^4 + 11*x^3 -18*x^2 + 12*x + 1$
(%i2) f2(x) := -4*x^3 + 28*x^2 -56*x + 32$
(%i3) (load(draw),load(qdraw) )$
                  qdraw(...), qdensity(...), syntax: type qdraw();

(%i4) qdraw(yr(-20,20),ex1(f1(x),x,-1,5,lc(blue) ),
                  ex1(f2(x),x,-1,5,lc(red)))$
```

which produces the plot

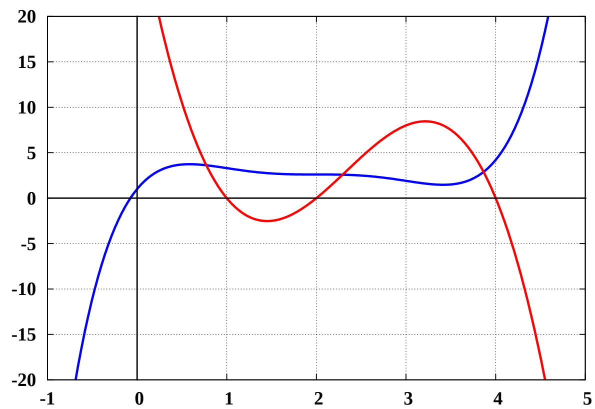

Figure 7: $f_1(x)$ (blue) and $f_2(x)$ (red)

Using the cursor on the plot, and working from left to right, f_1 becomes larger than f_2 at about $(0.76, 3.6)$, becomes less than f_2 at about $(2.3, 2.62)$, and becomes greater than f_2 at about $(3.86, 2.98)$. The **solve** function is not able to produce an analytic solution, but returns a polynomial whose roots are the solutions we want.

```
(%i5) solve(f1(x) = f2(x),x);
                        5        4         3         2
(%o5)          [0 = 3 x  - 30 x  + 150 x  - 460 x  + 680 x - 310]
(%i6) grind(%)$
[0 = 3*x^5-30*x^4+150*x^3-460*x^2+680*x-310]$
```

By selecting and copying the **grind(..)** output, we can use that result to paste in the definition of a function $p(x)$ which we can then use with `find_root`.

```
(%i7) p(x) := 3*x^5-30*x^4+150*x^3-460*x^2+680*x-310$
(%i8) x1 : find_root(p(x),x,.7,1);
(%o8)                         0.77205830452781
(%i9) x2 : find_root(p(x),x,2.2,2.4);
(%o9)                         2.291819210962957
(%i10) x3 : find_root(p(x),x,3.7,3.9);
(%o10)                         3.865127100061791
(%i11) map('p, [x1,x2,x3] );
(%o11)                 [0.0, 0.0, 9.0949470177292824E-13]
(%i12) [y1,y2,y3] : map('f1, [x1,x2,x3] );
(%o12)     [3.613992056691179, 2.575784006305792, 2.882949345140702]
```

We have checked the solutions by seeing how close to zero $p(x)$ is when x is one of the three roots $[x1, x2, x3]$. We now split up the integration into the two separate regions where one or the other function is larger.

```
(%i13) ratprint:false$
(%i14) i1 : integrate(f1(x) - f2(x),x,x1,x2);
                                  41875933
(%o14)                            --------
                                  7947418
(%i15) i2 : integrate(f2(x)-f1(x),x,x2,x3);
                                  12061231
(%o15)                            --------
                                  1741444
(%i16) area : i1 + i2;
                                 30432786985
(%o16)                           -----------
                                 2495489252
(%i17) area : float(area);
(%o17)                        12.19511843643877
```

Hence the total area enclosed is about 12.195. Maxima tries to calculate exactly, replacing floating point numbers with ratios of integers, and the default is to warn the user about these replacements. Hence we have used **`ratprint : false$`** to turn off this warning.

7.5 Arc Length of an Ellipse

Consider an ellipse whose equation is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{7.9}$$

in which we assume a is the semi-major axis of the ellipse so $a > b$. In the first quadrant ($0 \le x \le a$ and $0 \le y \le b$), we can solve for y as a function of x :

$$y(x) = b\sqrt{1 - (x/a)^2}. \tag{7.10}$$

If S is the arc length of the ellipse, then, by symmetry, one fourth of the arclength can be calculated using the first quadrant integral

$$\frac{S}{4} = \int_0^a \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx \tag{7.11}$$

We will start working with the argument of the square root, making a change of variable $x \to z$, with $x = a\, z$, so $dx = a\, dz$, and a will come outside the integral from the transformation of dx. The integration limits using the z variable are $0 \le z \le 1$.

We will then replace the semi-minor axis b by an expression depending on the ellipse eccentricity e, which has $0 < e \le 1$, and whose square is given by

$$e^2 = 1 - \left(\frac{b}{a}\right)^2 \le 1 \tag{7.12}$$

(since $b < a$), so

$$b = a\sqrt{1 - e^2} \tag{7.13}$$

```
(%i1) assume(a>0,b>0,a>b,e>0,e<1 )$
(%i2) y:b*sqrt(1 - (x/a)^2)$
(%i3) dydx : diff(y,x)$
```

```
(%i4) e1 : 1 + dydx^2;
                                    2  2
                                   b  x
(%o4)                           ----------- + 1
                                         2
                                 4      x
                                a  (1 - --)
                                         2
                                        a
(%i5) e2 : ( subst( [x = a*z,b = a*sqrt(1-e^2)],e1 ),ratsimp(%%) );
                                    2  2
                                   e  z  - 1
(%o5)                              ---------
                                     2
                                    z  - 1
(%i6) e3 : (-num(e2))/(-denom(e2));
                                        2  2
                                   1 - e  z
(%o6)                              ---------
                                         2
                                    1 - z
(%i7) e4 : dz*sqrt(num(e3))/sqrt(denom(e3));
                                          2  2
                              dz sqrt(1 - e  z )
(%o7)                         ------------------
                                          2
                                 sqrt(1 - z )
```

The two substitutions give us expression **e2**, and we use a desperate device to multiply the top and bottom by (-1) to get **e3**. We then ignore the factor a which comes outside the integral and consider what is now inside the integral sign (with the required square root).

We now make another change of variables, with $z \to u$, $z = \sin(u)$, so $dz = \cos(u)\,du$. The lower limit of integration $z = 0 = \sin(u)$ transforms into $u = 0$, and the upper limit $z = 1 = \sin(u)$ transforms into $u = \pi/2$.

```
(%i8) e5 : subst( [z = sin(u), dz = cos(u)*du ],e4 );
                                             2    2
                        du cos(u) sqrt(1 - e  sin (u))
(%o8)                   ------------------------------
                                            2
                              sqrt(1 - sin (u))
```

We now use **trigsimp** but help Maxima out with an **assume** statement about $\cos(u)$ and $\sin(u)$.

```
(%i9) assume(cos(u)>0, sin(u) >0)$
(%i10) e6 : trigsimp(e5);
                                        2    2
(%o10)                     du sqrt(1 - e  sin (u))
```

We then have

$$\frac{S}{4} = a \int_0^{\pi/2} \sqrt{1 - e^2 \sin^2 u}\, du \tag{7.14}$$

Although **integrate** cannot return an expression for this integral in terms of elementary functions, in this form one is able to recognise the standard trigonometric form of the complete elliptic integral of the second kind, (a function tabulated numerous places and also available via Maxima).

Let

$$\mathbf{E}(\mathbf{k}) = \mathbf{E}(\phi = \pi/2, \mathbf{k}) = \int_0^{\pi/2} \sqrt{1 - \mathbf{k}^2 \sin^2 \mathbf{u}}\, \mathbf{du} \tag{7.15}$$

be the definition of the complete elliptic integral of the second kind.

The "incomplete" elliptic integral of the second kind (with two arguments) is

$$\mathbf{E}(\phi, \mathbf{k}) = \int_0^{\phi} \sqrt{1 - \mathbf{k}^2 \sin^2 \mathbf{u}}\, \mathbf{du}. \tag{7.16}$$

Hence we have for the arc length of our ellipse

$$\mathbf{S} = 4\,\mathbf{a}\,\mathbf{E}(\mathbf{e}) = 4\,\mathbf{a}\,\mathbf{E}(\pi/2, \mathbf{e}) = 4\,\mathbf{a} \int_0^{\pi/2} \sqrt{1 - \mathbf{e}^2 \sin^2 \mathbf{u}}\, \mathbf{du} \tag{7.17}$$

We can evaluate numerical values using **elliptic_ec**, where
$\mathbf{E}(\mathbf{k}) = \mathbf{elliptic\_ec}(\mathbf{k}^2)$, so $\mathbf{S} = 4\,\mathbf{a}\,\mathbf{elliptic\_ec}(\mathbf{e}^2)$.

As a numerical example, take $\mathbf{a} = 3$, $\mathbf{b} = 2$, so $\mathbf{e}^2 = 5/9$, and

```
(%i11) float(12*elliptic_ec(5/9));
(%o11)                          15.86543958929059
```

We can check this numerical value using **`quad_qags`**:

```
(%i12) first( quad_qags(12*sqrt(1 - (5/9)*sin(u)^2),u,0,%pi/2) );
(%o12)                          15.86543958929059
```

7.6 Double Integrals and the Area of an Ellipse

Maxima has no core function which will compute a symbolic double definite integral, (although it would be easy to construct one). Instead of constructing such a homemade function, to evaluate the double integral

$$\int_{\mathbf{u1}}^{\mathbf{u2}} \mathbf{du} \int_{\mathbf{v1(u)}}^{\mathbf{v2(u)}} \mathbf{dv}\, \mathbf{f}(\mathbf{u}, \mathbf{v}) \equiv \int_{\mathbf{u1}}^{\mathbf{u2}} \left(\int_{\mathbf{v1(u)}}^{\mathbf{v2(u)}} \mathbf{f}(\mathbf{u}, \mathbf{v})\, \mathbf{dv} \right) \mathbf{du} \tag{7.18}$$

we use the Maxima code

```
        integrate( integrate( f(u,v),v,v1(u),v2(u) ), u, u1,u2 )
```

in which $\mathbf{f}(\mathbf{u}, \mathbf{v})$ can either be an expression depending on the variables $\mathbf{u}$ and $\mathbf{v}$, or a Maxima function, and likewise $\mathbf{v1}(\mathbf{u})$ and $\mathbf{v2}(\mathbf{u})$ can either be expressions depending on $\mathbf{u}$ or Maxima functions. Both $\mathbf{u}$ and $\mathbf{v}$ are "dummy variables", since the value of the resulting double integral does not depend on our choice of symbols for the integration variables; we could just as well use $\mathbf{x}$ and $\mathbf{y}$.

Example 1: Area of a Unit Square

The area of a unit square (the sides have length $\mathbf{1}$) is:

$$\int_0^1 \mathbf{dx} \int_0^1 \mathbf{dy} \equiv \int_0^1 \left(\int_0^1 \mathbf{dy} \right) \mathbf{dx} \tag{7.19}$$

which is done in Maxima as:

```
(%i1) integrate( integrate(1,y,0,1), x,0,1 );
(%o1)                              1
```

Example 2: Area of an Ellipse

We seek the area of the ellipse such that points $(\mathbf{x}, \mathbf{y})$ on the boundary must satisfy:

$$\frac{\mathbf{x}^2}{\mathbf{a}^2} + \frac{\mathbf{y}^2}{\mathbf{b}^2} = \mathbf{1}. \tag{7.20}$$

The area will be four times the area of the first quadrant. The "first quadrant" refers to the region $\mathbf{0} \leq \mathbf{x} \leq \mathbf{a}$ and $\mathbf{0} \leq \mathbf{y} \leq \mathbf{b}$. For a given value of $\mathbf{y} > \mathbf{0}$, the region inside the arc of the ellipse in the first quadrant is determined by $\mathbf{0} \leq \mathbf{x} \leq \mathbf{x_{max}}$, where $\mathbf{x_{max}} = (\mathbf{a}/\mathbf{b})\sqrt{\mathbf{b}^2 - \mathbf{y}^2}$. For a given value of $\mathbf{x} > \mathbf{0}$, the region inside the arc of the ellipse in the first quadrant is determined by $\mathbf{0} \leq \mathbf{y} \leq \mathbf{y_{max}}$, where $\mathbf{y_{max}} = (\mathbf{b}/\mathbf{a})\sqrt{\mathbf{a}^2 - \mathbf{x}^2}$.

One way to find the area of the first quadrant of this ellipse is to sum the values of $\mathbf{dA} = \mathbf{dx}\,\mathbf{dy}$ by "rows", fixing $\mathbf{y}$ and summing over $\mathbf{x}$ from $\mathbf{0}$ to $\mathbf{x_{max}}$ (which depends on the $\mathbf{y}$ chosen). That first sum over the $\mathbf{x}$ coordinate accumulates the area of that row, located at $\mathbf{y}$ and having width $\mathbf{dy}$. To get the total area (of the first quadrant) we then sum over rows, by letting $\mathbf{y}$ vary from $\mathbf{0}$ to $\mathbf{b}$.
This method corresponds to the formula

$$\frac{\mathbf{A}}{\mathbf{4}} = \int_0^{\mathbf{b}} \left(\int_0^{(\mathbf{a}/\mathbf{b})\sqrt{\mathbf{b}^2 - \mathbf{y}^2}} \mathbf{dx} \right) \mathbf{dy}. \tag{7.21}$$

Here we calculate the first quadrant area using this method of summing over the area of each row.

```
(%i1) facts();
(%o1)                             []
(%i2) assume(a > 0, b > 0, x > 0, x < a, y > 0,y < b )$
(%i3) facts();
(%o3)            [a > 0, b > 0, x > 0, a > x, y > 0, b > y]
(%i4) [xmax : (a/b)*sqrt(b^2 - y^2),ymax : (b/a)*sqrt(a^2-x^2)]$
(%i5) integrate( integrate( 1,x,0,xmax), y,0,b );
                                   %pi a b
(%o5)                              -------
                                      4
```

which implies that the area interior to the complete ellipse is $\pi\,\mathbf{a}\,\mathbf{b}$.

Note that we have tried to be "overly helpful" to Maxima's **integrate** function by constructing an "assume list" with everything we can think of about the variables and parameters in this problem. The main advantage of doing this is to reduce the number of questions which **integrate** decides it has to ask the user.

You might think that **integrate** would not ask for the sign of $(y - b)$, or the sign of $(x - a)$, since it should infer that sign from the integration limits. However, the **integrate** algorithm is "super cautious" in trying to never present you with a wrong answer. The general philosophy is that the user should be willing to work with **integrate** to assure a correct answer, and if that involves answering questions, then so be it.

The second way to find the area of the first quadrant of this ellipse is to sum the values of $dA = dx\,dy$ by "columns", fixing x and summing over y from 0 to y_{max} (which depends on the x chosen). That first sum over the y coordinate accumulates the area of that column, located at x and having width dx. To get the total area (of the first quadrant) we then sum over columns, by letting x vary from 0 to a.

This method corresponds to the formula

$$\frac{A}{4} = \int_0^a \left(\int_0^{(b/a)\sqrt{a^2-x^2}} dy \right) dx \tag{7.22}$$

and is implemented by

```
(%i6) integrate( integrate( 1,y,0,ymax), x,0,a );
                                              %pi a b
(%o6)                                         -------
                                                 4
```

Example 3: Moment of Inertia for Rotation about the x-axis

We next calculate the moment of inertia for rotation of an elliptical lamina (having semi-axes a, b) about the x axis. We will call this quantity I_x. Each small element of area $dA = dx\,dy$ has a mass dm given by $\sigma\,dA$, where σ is the mass per unit area, which we assume is a constant independent of where we are on the lamina, and which we can express in terms of the total mass m of the elliptical lamina, and the total area $A = \pi\,a\,b$ as

$$\sigma = \frac{\text{total mass}}{\text{total area}} = \frac{m}{\pi\,a\,b} \tag{7.23}$$

Each element of mass dm contributes an amount $y^2\,dm$ to the moment of inertia I_x, where y is the distance of the mass element from the x axis. The complete value of I_x is then found by summing this quantity over the whole elliptical laminate, or because of the symmetry, by summing this quantity over the first quadrant and multiplying by 4.

$$I_x = \int\int_{\text{ellipse}} y^2\,dm = \int\int_{\text{ellipse}} y^2\,\sigma\,dx\,dy = 4\,\sigma\int\int_{\text{first quadrant}} y^2\,dx\,dy \tag{7.24}$$

Here we use the method of summing over rows:

```
(%i7) sigma : m/(%pi*a*b)$
(%i8) 4*sigma*integrate( integrate(y^2,x,0,xmax),y,0,b );
                                                2
                                               b  m
(%o8)                                          ----
                                                4
```

Hence we have derived $I_x = m\,b^2/4$ for the moment of inertia of an elliptical laminate having semi-axes a, b for rotation about the x axis.

Finally, let's use **forget** to remove our list of assumptions, to show you the types of questions which can arise. We try calculating the area of the first quadrant, using the method of summing over rows, without any assumptions provided:

```
(%i9) forget(a > 0, b > 0, x > 0, x < a, y > 0,y < b )$
(%i10) facts();
(%o10)                           []
(%i11) integrate( integrate(1,x,0,xmax),y,0,b);
Is  (y - b) (y + b)  positive, negative, or zero?
n;
     2    2
Is  b   - y   positive or zero?
p;
Is  b  positive, negative, or zero?
p;
                                        %pi a b
(%o11)                                  -------
                                           4
```

and if we just tell Maxima that **b** is positive:

```
(%i12) assume(b>0)$
(%i13) integrate( integrate(1,x,0,xmax),y,0,b);
Is  (y - b) (y + b)  positive, negative, or zero?
n;
     2    2
Is  b   - y   positive or zero?
p;
                                        %pi a b
(%o13)                                  -------
                                           4
```

This may give you some feeling for the value of providing some help to **integrate**.

7.7 Triple Integrals: Volume and Moment of Inertia of a Solid Ellipsoid

Consider an ellipsoid with semi-axes (a, b, c) such that $-a \le x \le a$, $-b \le y \le b$, and $-c \le z \le c$. We also assume here that $a > b > c$. Points (x, y, z) which live on the surface of this ellipsoid satisfy the equation of the surface

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1. \tag{7.25}$$

Volume

The volume of this ellipsoid will be 8 times the volume of the first octant, defined by $0 \le x \le a$, $0 \le y \le b$, and $0 \le z \le c$. To determine the volume of the first octant we pick fixed values of (x, y) somewhere in the first octant, and sum the elementary volume $dV(x, y, z) = dx\, dy\, dz$ over all accessible values of z in this first octant, $0 \le z \le c\sqrt{1 - (x/a)^2 - (y/b)^2}$. The result will still be proportional to $dx\, dy$, and we sum the z-direction cylinders over the entire range of y (accessible in the first octant), again holding x fixed as before, so for given x, we have $0 \le y \le b\sqrt{1 - (x/a)^2}$. We now have a result proportional to dx (a sheet of thickness dx whose normal is in the direction of the x axis) which we sum over all values of x accessible in the first octant (with no restrictions): $0 \le x \le a$. Thus we need to evaluate the triple integral

$$\frac{V}{8} = \int_0^a dx \int_0^{ymax(x)} dy \int_0^{zmax(x,y)} dz \equiv \int_0^a \left[\int_0^{ymax(x)} \left(\int_0^{zmax(x,y)} dz \right) dy \right] dx \tag{7.26}$$

Here we call on **integrate** to do these integrals.

```
(%i1) assume(a>0,b>0,c>0,a>b,a>c,b>c)$
(%i2) assume(x>0,x<a,y>0,y<b,z>0,z<c)$
(%i3) zmax:c*sqrt(1-x^2/a^2-y^2/b^2)$
(%i4) ymax:b*sqrt(1-x^2/a^2)$
(%i5) integrate( integrate( integrate(1,z,0,zmax),y,0,ymax),x,0,a );
     2  2    2  2    2  2
Is  a  y  + b  x  - a  b   positive, negative, or zero?

n;
                                   %pi a b c
(%o5)                              ---------
                                       6
(%i6) vol : 8*%;
                                  4 %pi a b c
(%o6)                             -----------
                                       3
```

To answer the sign question posed by Maxima, we can look at the special case $y = 0$ and note that since $x^2 \le a^2$, the expression will almost always be negative (ie., except for points of zero measure). Hence the expression is negative for all points interior to the surface of the ellipsoid. We thus have the result that the volume of an ellipsoid having semi-axes (a, b, c) is given by

$$V = \frac{4\,\pi}{3} a\,b\,c \tag{7.27}$$

We can now remove the assumption $a > b > c$, since what we call the x axis is up to us and we could have chosen any of the principal axis directions of the ellipsoid the x axis.

Of course we will get the correct answer if we integrate over the total volume:

```
(%i7) [zmin:-zmax,ymin:-ymax]$
(%i8) integrate( integrate( integrate(1,z,zmin,zmax),y,ymin,ymax),x,-a,a );
     2  2    2  2    2  2
Is  a  y  + b  x  - a  b   positive, negative, or zero?

n;
                                  4 %pi a b c
(%o8)                             -----------
                                       3
```

Moment of Inertia

If each small volume element $dV = dx\,dy\,dz$ has a mass given by $dm = \rho\,dV$, where ρ is the mass density, and if ρ is a constant, then it is easy to calculate the moment of inertia I_3 for rotation of the ellipsoid about the z axis:

$$I_3 = \int\int\int (x^2 + y^2)\,dm = \rho \int\int\int (x^2 + y^2)\,dx\,dy\,dz \tag{7.28}$$

where the integration is over the volume of the ellipsoid. The constant mass density ρ is the mass of the ellipsoid divided by its volume.

Here we define that constant density in terms of our previously found volume **vol** and the mass **m**, and proceed to calculate the moment of inertia:

```
(%i9) rho:m/vol;
                                          3 m
(%o9)                                 -----------
                                      4 %pi a b c
(%i10) i3:rho*integrate(integrate(integrate(x^2+y^2,z,zmin,zmax),
                       y,ymin,ymax),x,-a,a );
    2   2     2   2     2   2
Is   a   y   + b   x   - a   b    positive, negative, or zero?

n;
                                     5   3           7
                             (8 %pi a   b   + 8 %pi a   b) m
(%o10)                       ----------------------------
                                            5
                                     40 %pi a   b
(%i11) ratsimp(i3);
                                        2     2
                                      (b   + a ) m
(%o11)                                -----------
                                           5
```

Hence the moment of inertia for the rotation about the z axis of a solid ellipsoid having mass m, and semi-axes (a, b, c) is:

$$I_3 = m\,(a^2 + b^2)/5. \tag{7.29}$$

7.8 Derivative of a Definite Integral with Respect to a Parameter

Consider a definite integral in which the dummy integration variable is x and the integrand $f(x, y)$ is a function of both x and some parameter y. We also assume that the limits of integration (a, b) are also possibly functions of the parameter y. You can find the proof of the following result (which assumes some mild restrictions on the funtions $f(x, y)$, $a(y)$ and $b(y)$) in calculus texts:

$$\frac{d}{dy}\int_{a(y)}^{b(y)} f(x, y)\,dx = \int_a^b \frac{\partial f(x, y)}{\partial y}\,dy + f(b(y), y)\,\frac{d\,b}{d\,y} - f(a(y), y)\,\frac{d\,a}{d\,y} \tag{7.30}$$

Here we ask Maxima for this result for arbitrary functions:

```
(%i1) expr : 'integrate(f(x,y),x,a(y),b(y) );
                                b(y)
                               /
                               [
(%o1)                          I     f(x, y) dx
                               ]
                               /
                                a(y)
(%i2) diff(expr,y);
                                                          b(y)
                                                         /
                  d                          d           [       d
(%o2) f(b(y), y) (-- (b(y))) - f(a(y), y) (-- (a(y))) + I      (-- (f(x, y))) dx
                  dy                         dy          ]       dy
                                                         /
                                                          a(y)
```

and we see that Maxima assumes we have enough smoothness in the functions involved to write down the formal answer in the correct form.

Example 1

We next display a simple example and begin with the simplest case, which is that the limits of integration do not depend on the parameter y.

```
(%i3) expr : 'integrate(x^2 + 2*x*y, x,a,b);
                               b
                              /
                              [              2
(%o3)                         I   (2 x y + x ) dx
                              ]
                              /
                               a
(%i4) diff(expr,y);
                                      b
                                     /
                                     [
(%o4)                              2 I   x dx
                                     ]
                                     /
                                      a
(%i5) (ev(%,nouns), ratsimp(%%) );
                                     2    2
(%o5)                               b  - a
(%i6) ev(expr,nouns);
                              2      3      2      3
                           3 b  y + b    3 a  y + a
(%o6)                      ----------- - -----------
                                3             3
(%i7) diff(%,y);
                                     2    2
(%o7)                               b  - a
```

In the last two steps, we have verified the result by first doing the original integral and then taking the derivative.

Example 2

As a second example, we use an arbitrary upper limit $b(y)$, and then evaluate the resulting derivative expression for $b(y) = y^2$.

```
(%i1) expr : 'integrate(x^2 + 2*x*y, x,a,b(y) );
                              b(y)
                             /
                             [                2
(%o1)                        I      (2 x y + x ) dx
                             ]
                             /
                              a
```

```
(%i2) diff(expr,y);
                                                    b(y)
                                                   /
                 2                 d               [
(%o2)         (b (y) + 2 y b(y)) (-- (b(y))) + 2 I     x dx
                                  dy               ]
                                                   /
                                                    a
(%i3) ( ev(%,nouns,b(y)=y^2 ), expand(%%) );
                                   5     4    2
(%o3)                           2 y  + 5 y  - a
(%i4) ev(expr,nouns);
                         3          2         2      3
                        b (y) + 3 y b (y)   3 a  y + a
(%o4)                   ----------------- - -----------
                                3                3
(%i5) diff(%,y);
           2     d                       d                2
        3 b (y) (-- (b(y))) + 6 y b(y) (-- (b(y))) + 3 b (y)
                 dy                      dy                       2
(%o5)   --------------------------------------------------- - a
                                   3
(%i6) ( ev(%,nouns,b(y)=y^2), (expand(%%) );
                                   5     4    2
(%o6)                           2 y  + 5 y  - a
```

We have again done the differentiation two ways as a check on consistency.

Example 3

An easy example is to derive

$$\frac{\mathrm{d}}{\mathrm{d}\,\mathrm{t}}\int_{\mathrm{t}}^{\mathrm{t}^2}(2\,\mathrm{x}+\mathrm{t})\,\mathrm{d}\,\mathrm{x} = 4\,\mathrm{t}^3 + 3\,\mathrm{t}^2 - 4\,\mathrm{t} \tag{7.31}$$

```
(%i7) expr : 'integrate(2*x + t,x,t,t^2);
                                  2
                                 t
                                /
                                [
(%o7)                           I    (2 x + t) dx
                                ]
                                /
                                 t
(%i8) (diff(expr,t),expand(%%) );
                                  3      2
(%o8)                          4 t  + 3 t  - 4 t
```

Example 4

Here is an example which shows a common use of the differentiation of an integral with respect to a parameter. The integral

$$f_1(a,w) = \int_0^{\infty} x\,e^{-a\,x}\cos(w\,x)\,dx \tag{7.32}$$

can be done by Maxima with no questions asked, if we tell Maxima that $a > 0$ and $w > 0$.

```
(%i1) assume( a > 0, w > 0 )$
(%i2) integrate(x*exp(-a*x)*cos(w*x),x,0,inf);
                                          2    2
                                         w  - a
(%o2)                            - -----------------
                                     4      2  2    4
                                    w  + 2 a  w  + a
```

But if we could not find this result directly as above, we could find the result by setting $f_1(a,w) = -\partial f_2(a,w)/(\partial a)$, where

$$f_2(a,w) = \int_0^{\infty} e^{-a\,x}\cos(w\,x)\,dx \tag{7.33}$$

since the differentiation will result in the integrand being multiplied by the factor $(-x)$ and produce the negative of the integral of interest.

```
(%i3) i1 : 'integrate(exp(-a*x)*cos(w*x),x,0,inf)$
(%i4) i2 : ev(i1,nouns);
                                           a
(%o4)                                   -------
                                         2    2
                                        w  + a
(%i5) di2 : (diff(i2,a), ratsimp(%%) );
                                         2    2
                                        w  - a
(%o5)                             -----------------
                                    4      2  2    4
                                   w  + 2 a  w  + a
(%i6) result : (-1)*(diff(i1,a) = di2 );
                inf
               /                                    2    2
               [        - a x                      w  - a
(%o6)          I     x %e      cos(w x) dx = - -----------------
               ]                                 4      2  2    4
               /                                w  + 2 a  w  + a
                0
```

which reproduces the original result returned by **integrate**.

7.9 Integration by Parts

Suppose $\mathbf{f(x)}$ and $\mathbf{g(x)}$ are two continuously differentiable functions in the interval of interest. Then the **integration by parts rule** states that given an interval with endpoints $(\mathbf{a}, \mathbf{b})$, (and of course assuming the derivatives exist) one has

$$\int_{\mathbf{a}}^{\mathbf{b}} \mathbf{f(x)\, g'(x)\, dx} = [\mathbf{f(x)\, g(x)}]_{\mathbf{a}}^{\mathbf{b}} - \int_{\mathbf{a}}^{\mathbf{b}} \mathbf{f'(x)\, g(x)\, dx} \tag{7.34}$$

where the prime indicates differentiation. This result follows from the product rule of differentiation. This rule is often stated in the context of indefinite integrals as

$$\int \mathbf{f(x)\, g'(x)\, dx} = \mathbf{f(x)\, g(x)} - \int \mathbf{f'(x)\, g(x)\, dx} \tag{7.35}$$

or in an even shorter form, with $\mathbf{u} = \mathbf{f(x)}$, $\mathbf{du} = \mathbf{f'(x)\, dx}$, $\mathbf{v} = \mathbf{g(x)}$, and $\mathbf{dv} = \mathbf{g'(x)\, dx}$, as

$$\int \mathbf{u\, dv} = \mathbf{u\, v} - \int \mathbf{v\, du} \tag{7.36}$$

In practice, we are confronted with an integral whose integrand can be viewed as the product of two factors, which we will call $\mathbf{f(x)}$ and $\mathbf{h(x)}$:

$$\int \mathbf{f(x)\, h(x)\, dx} \tag{7.37}$$

and we wish to use integration by parts to get an integral involving the derivative of the first factor, $\mathbf{f(x)}$, which will hopefully result in a simpler integral. We then identify $\mathbf{h(x)} = \mathbf{g'(x)}$ and solve this equation for $\mathbf{g(x)}$ (by integrating; this is also a choice based on the ease of integrating the second factor $\mathbf{h(x)}$ in the given integral). Having $\mathbf{g(x)}$ in hand we can then write out the result using the indefinite integral integration by parts rule above. We can formalize this process for an *indefinite integral* with the Maxima code:

```
(%i1) iparts(f,h,var):= block([g ],
         g : integrate(h,var),
         f*g - 'integrate(g*diff(f,var),var ) )$
```

Let's practice with the integral $\int \mathbf{x^2 \sin(x)\, dx}$, in which $\mathbf{f(x)} = \mathbf{x^2}$ and $\mathbf{h(x)} = \mathbf{\sin(x)}$, so we need to be able to integrate $\sin(x)$ and want to transfer a derivative on to $\mathbf{x^2}$, which will reduce the first factor to $\mathbf{2\,x}$. Notice that it is usually easier to work with "Maxima expressions" rather than with "Maxima functions" in a problem like this.

```
(%i2) iparts(x^2,sin(x),x);
                                 /
                                 [                2
(%o2)                       2 I x cos(x) dx - x  cos(x)
                                 ]
                                 /
(%i3) (ev(%,nouns),expand(%%) );
                                             2
(%o3)                    2 x sin(x) - x  cos(x) + 2 cos(x)
(%i4) collectterms(%,cos(x) );
                                                   2
(%o4)                      2 x sin(x) + (2 - x ) cos(x)
```

If we were not using Maxima, but doing everything by hand, we would use two integrations by parts (in succession) to remove the factor $\mathbf{x^2}$ entirely, reducing the original problem to simply knowing the integrals of $\sin(x)$ and $\cos(x)$.

Of course, with an integral as simple as this example, there is no need to help Maxima out by integrating by parts.

```
(%i5) integrate(x^2*sin(x),x);
                                              2
(%o5)                   2 x sin(x) + (2 - x ) cos(x)
```

We can write a similar Maxima function to transform definite integrals via integration by parts.

```
(%i6) idefparts(f,h,var,v1,v2):= block([g ],
         g : integrate(h,var),
           'subst(v2,var, f*g)  - 'subst(v1,var, f*g ) -
               'integrate(g*diff(f,var),var,v1,v2 ) )$
(%i7) idefparts(x^2,sin(x),x,0,1);
          1
         /
         [                                    2
(%o7) 2 I   x cos(x) dx + substitute(1, x, - x   cos(x))
         ]
         /
          0
                                                                      2
                                                - substitute(0, x, - x   cos(x))
(%i8) (ev(%,nouns),expand(%%) );
(%o8)                       2 sin(1) + cos(1) - 2
(%i9) integrate(x^2*sin(x),x,0,1);
(%o9)                       2 sin(1) + cos(1) - 2
```

7.10 Change of Variable and changevar

Many integrals can be evaluated most easily by making a change of variable of integration. A simple example is:

$$\int 2\,x\,(x^2+1)^3\,dx = \int (x^2+1)^3\,d\,(x^2+1) = \int u^3\,du = u^4/4 = (x^2+1)^4/4 \tag{7.38}$$

There is a function in Maxima, called **changevar** which will help you change variables in a one-dimensional integral (either indefinite or definite). However, this function is buggy at present and it is safer to do the change of variables "by hand".

> Function: **changevar(integral-expression, g(x,u), u, x)**
> Makes the change of variable in a given *noun form* **integrate** expression such that the old variable of integration is x, the new variable of integration is u, and x and u are related by the equation $g(x,u) = 0$.

Example 1

Here we use this Maxima function on the simple indefinite integral $\int 2\,x\,(x^2+1)^3\,dx$ we have just done "by hand":

```
(%i1) expr : 'integrate(2*x*(x^2+1)^3,x);
                                /
                                [      2     3
(%o1)                         2 I x (x  + 1)  dx
                                ]
                                /
(%i2) changevar(expr,x^2+1-u,u,x);
                                    /
                                    [  3
(%o2)                               I u  du
                                    ]
                                    /
(%i3) ev(%, nouns);
                                      4
                                     u
(%o3)                                --
                                     4
(%i4) ratsubst(x^2+1,u,%);
                           8      6      4      2
                          x  + 4 x  + 6 x  + 4 x  + 1
(%o4)                     ---------------------------
                                      4
(%i5) subst(x^2+1,u,%o3);
                                    2     4
                                  (x  + 1)
(%o5)                             ---------
                                      4
(%i6) subst(u=(x^2+1),%o3);
                                    2     4
                                  (x  + 1)
(%o6)                             ---------
                                      4
```

The original indefinite integral is a function of x, which we obtain by replacing u by its equivalent as a function of x. We have shown three ways to make this replacement to get a function of x, using **subst** and **ratsubst**.

Example 2

As a second example, we use a change of variables to find $\int \left[(x+2)/\sqrt{x+1}\right]\,dx$.

```
(%i7) expr : 'integrate( (x+2)/sqrt(x+1),x);
                               /
                               [    x + 2
(%o7)                          I ----------- dx
                               ] sqrt(x + 1)
                               /
```

```
(%i8) changevar(expr,u - sqrt(x+1),u,x);
Is  u  positive, negative, or zero?

pos;
                                 /
                                 [      2
(%o8)                            I  (2 u  + 2) du
                                 ]
                                 /
(%i9) ev(%, nouns);
                                     3
                                  2 u
(%o9)                             ---- + 2 u
                                   3
(%i10) subst(u = sqrt(x+1),%);
                                  3/2
                         2 (x + 1)
(%o10)                   ------------ + 2 sqrt(x + 1)
                              3
```

Of course Maxima can perform the original integral in these two examples without any help, as in:

```
(%i11) integrate((x+2)/sqrt(x+1),x);
                                   3/2
                            (x + 1)
(%o11)                   2 (---------- + sqrt(x + 1))
                                3
```

However, there are occasionally cases in which you can help Maxima find an integral (for which Maxima can only return the noun form) by first making your own change of variables and then letting Maxima try again.

Example 3

Here we use **changevar** with a definite integral, using the same integrand as in the previous example. For a definite integral, the variable of integration is a "dummy variable", and the result is not a function of that dummy variable, so there is no issue about replacing the new integration variable **u** by the original variable **x** in the result.

```
(%i12) expr : 'integrate( (x+2)/sqrt(x+1),x,0,1);
                                 1
                                /
                                [      x + 2
(%o12)                          I   ----------- dx
                                ]   sqrt(x + 1)
                                /
                                 0
```

```
(%i13) changevar(expr,u - sqrt(x+1),u,x);
Is  u  positive, negative, or zero?

pos;
                             sqrt(2)
                            /
                            [           2
(%o13)                      I       (2 u  + 2) du
                            ]
                            /
                             1
(%i14) ev(%, nouns);
                                10 sqrt(2)   8
(%o14)                          ---------- - -
                                    3        3
```

Example 4

We next discuss an example which shows that one needs to pay attention to the possible introduction of obvious sign errors when using **changevar**. The example is the evaluation of the definite integral $\int_0^1 e^{y^2}\, dy$, in which y is a real variable. Since the integrand is a positive (real) number over the interval $0 < y \leq 1$, the definite integral must be a positive (real) number. The answer returned directly by Maxima's **integrate** function is:

```
(%i1) fpprintprec:8$
(%i2) i1:integrate (exp (y^2),y,0,1);
                              sqrt(%pi) %i erf(%i)
(%o2)                       - --------------------
                                       2
```

and **float** yields

```
(%i3) float(i1);
(%o3)                         - 0.886227 %i erf(%i)
```

so we see if we can get a numerical value from **`erf(%i)`** by multiplying **`%i`** by a floating point number:

```
(%i4) erf(1.0*%i);
(%o4)                             1.6504258 %i
```

so to get a numerical value for the integral we use the same trick

```
(%i5) float(subst(1.0*%i,%i,i1));
(%o5)                               1.4626517
```

Maxima's symbol **erf(z)** represents the error function $\mathrm{Erf}(z)$. We have discussed the Maxima function **erf(x)** for real x in Example 4 in Sec.(7.2). Here we have a definite integral result returned in terms of **`erf(%i)`**, which is the error function with a pure imaginary agument and we have just seen that **`erf(%i)`** is purely imaginary with an approximate value **`1.65*%i`**.

We can confirm the numerical value of the integral **`i1`** using the quadrature routine **`quad_qags`**:

```
(%i6) quad_qags(exp(y^2),y,0,1);
(%o6)                 [1.4626517, 1.62386965E-14, 21, 0]
```

and we see agreement.

Let's ask Maxima to change variables in this definite integral from $\mathbf{y}$ to $\mathbf{u} = \mathbf{y^2}$ in the following way:

```
(%i7) expr : 'integrate(exp(y^2),y,0,1);
                                   1
                                  /     2
                                  [    y
(%o7)                             I  %e    dy
                                  ]
                                  /
                                   0
(%i8) changevar(expr,y^2-u,u,y);
                                   1
                                  /      u
                                  [    %e
                                  I  ------- du
                                  ]  sqrt(u)
                                  /
                                   0
(%o8)                           - -------------
                                        2
(%i9) ev(%, nouns);
                             sqrt(%pi) %i erf(%i)
(%o9)                        --------------------
                                      2
(%i10) float(subst(1.0*%i,%i,%));
(%o10)                            - 1.4626517
```

which is the negative of the correct result. Evidently, Maxima uses **solve**$(\mathbf{y^2} = \mathbf{u},\ \mathbf{y})$ to find $\mathbf{y}(\mathbf{u})$ and even though there are two solutions $\mathbf{y} = \pm\sqrt{\mathbf{u}}$, Maxima picks the wrong solution without asking the user a clarifying question. **We need to force Maxima to use the correct relation** between $\mathbf{y}$ and $\mathbf{u}$, as in:

```
(%i11) changevar(expr,y-sqrt(u),u,y);
Is  y  positive, negative, or zero?

pos;
                                  1
                                 /      u
                                 [    %e
                                 I  ------- du
                                 ]  sqrt(u)
                                 /
                                  0
(%o11)                           -------------
                                       2
(%i12) ev(%, nouns);
                              sqrt(%pi) %i erf(%i)
(%o12)                      - --------------------
                                       2
(%i13) float(subst(1.0*%i,%i,%));
(%o13)                             1.4626517
```

which is now the correct result with the correct sign.

Example 5

We now discuss an example of a change of variable in which **changevar** produces the wrong overall sign, even though we try to be very careful. We consider the indefinite integral $\int \left(x/\sqrt{x^2-4} \right) dx$, which **integrate** returns as:

```
(%i1) integrate(x/sqrt(x^2-4),x);
                                        2
(%o1)                           sqrt(x   - 4)
```

Now consider the change of variable $x \to t$ with $x = 2/\cos(t)$.

We will show first the **changevar** route (with its error) and then how to do the change of variables "by hand", but with Maxima'a assistance. Here we begin with assumptions about the variables involved.

```
(%i2) assume(x > 2, t > 0, t < 1.5, cos(t) > 0, sin(t) > 0 );
(%o2)          [x > 2, t > 0, t < 1.5, cos(t) > 0, sin(t) > 0]
(%i3) nix : 'integrate(x/sqrt(x^2-4),x);
                              /
                              [      x
(%o3)                         I ------------ dx
                              ]       2
                              / sqrt(x   - 4)
(%i4) nixt : changevar(nix,x-2/cos(t), t, x) ;
                   /
                   [                  sin(t)
(%o4)        - 2 %i I ----------------------------------------- dt
                   ]                     2
                   / sqrt(cos(t) - 1) cos (t) sqrt(cos(t) + 1)
(%i5) nixt : rootscontract(nixt);
                           /
                           [          sin(t)
(%o5)                 - 2 %i I ------------------------- dt
                           ]    2             2
                           / cos (t) sqrt(cos (t) - 1)
(%i6) nixt : scanmap('trigsimp,nixt);
                                  /
                                  [    1
(%o6)                         - 2 I ------- dt
                                  ]    2
                                  / cos (t)
(%i7) ev(nixt,nouns);
(%o7)                           - 2 tan(t)
```

Since we have assumed $t > 0$, we have $\tan(t) > 0$, so **changevar** is telling us the indefinite integral is a negative number for the range of t assumed.

Since we are asking for an indefinite integral, and we want the result in terms of the original variable x, we would need to do some more work on this answer, maintaing the assumptions we have made. We will do that work after we have repeated this change of variable, doing it "by hand".

We work on the product $f(x)\,dx$:

```
(%i8) ix : subst(x=2/cos(t),x/sqrt(x^2 - 4) )* diff(2/cos(t));
                                4 sin(t) del(t)
(%o8)                      -------------------------
                                  4            3
                           sqrt(------- - 4) cos (t)
                                  2
                                cos (t)
(%i9) ix : trigsimp(ix);
                                  2 del(t)
(%o9)                           - -----------
                                     2
                                  sin (t) - 1
(%i10) ix : ratsubst(1,cos(t)^2+sin(t)^2,ix);
                                  2 del(t)
(%o10)                             --------
                                     2
                                  cos (t)
(%i11) integrate(coeff(ix,del(t) ) ,t);
(%o11)                            2 tan(t)
```

which is the result **changevar** should have produced. Now let's show how we can get back to the indefinite integral produced the direct use of **integrate**.

```
(%i12) subst(tan(t)= sqrt(sec(t)^2-1),2*tan(t) );
                                       2
(%o12)                        2 sqrt(sec (t) - 1)
(%i13) subst(sec(t)=1/cos(t),%);
                                       1
(%o13)                        2 sqrt(------- - 1)
                                       2
                                     cos (t)
(%i14) subst(cos(t)=2/x,%);
                                       2
                                      x
(%o14)                          2 sqrt(-- - 1)
                                      4

(%i15) ratsimp(%);
                                      2
(%o15)                           sqrt(x  - 4)
```

This concludes our discussion of a change of variable of integration and our discussion of symbolic integration.