# Numerical Integration Tool

## Introduction

This application uses the *Double Exponential* [1] numerical integration method to evaluate integrals numerically. It also performs additional related calculations and can iterate through sets of input values to save the user from manually setting individual input values.

The Double Exponential Transform, also known as the Sinh-Tanh Transform, is very efficient at evaluating integrals where the integrand value is not cyclical. It may not be appropriate for cyclical functions such as (sin x)/x.

The mathematical operations are performed to an arbitrary precision which may be set by the user. The default precision is about 24 decimal digits, but can be increased or decreased by the user as necessary. The extended precision math library that is used in this application was developed by Robert Delaney [2]. In addition to the usual scientific functions, the library includes a number of specialized scientific functions. All of the operators and functions available for use in this application are listed in **Appendix A**.

## Quick Start Guide

The best way to introduce the application's capabilities is with some simple examples. When the application starts up, the following window is displayed.

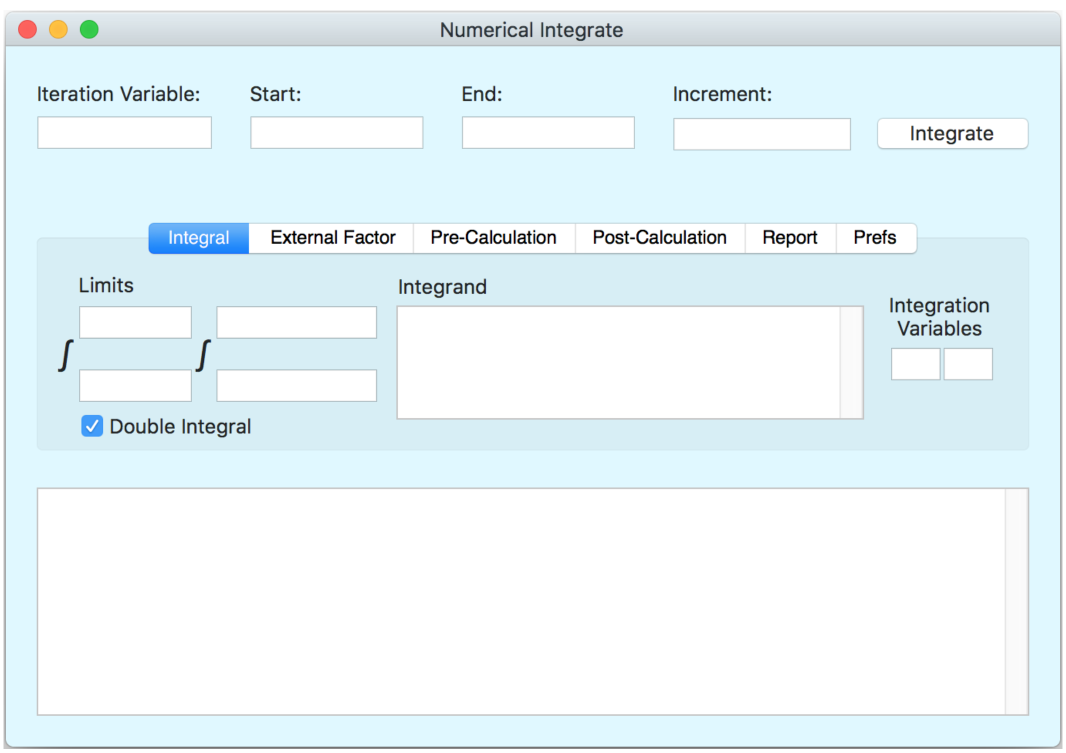


*Figure 1*

In the centre of the window is a tab panel that shows the Integral tab. This is where the user will enter most of the important information related to the integral to be be evaluated. The data entry fields are positioned where one would expect to see them when manually writing out the integral. Clicking the **Double Integral** check box switches between single and double integration modes, and the entry fields for the outer integral's limits and variable of integration are displayed or hidden accordingly.

Calculation output appears in the text field at the bottom of the window. The eight data entry fields at the top of the window, as well as the other tabs in the central tab panel will be described later. But first, several short examples will be given.

For these examples we will use functions whose integrals have analytical values that we can use for comparison.

**Example 1 - Single Integral**

The function $y = \sqrt{(1-x^2)}$ describes a semicircular line of radius = 1, with the end points sitting on the $x$ axis. If we integrate this with respect to $x$ over the limits –1 to +1, we will get the area of the semicircle which is equal to $\frac{1}{2}\pi r^2 = \frac{\pi}{2}$. If we include a factor of two in the integrand, then the result should be equal to $\pi$. Thus, we have this integral:

$$\int_{-1}^{1} 2\sqrt{1-x^2}\, dx$$

In computer notation, the integrand is: `2*sqrt(1-x^2)`
To try this example, do the following:

- Copy the above integrand expression and paste it into the Integrand text field;
- Set the lower and upper limits are set to `-1` and `1` respectively;
- Enter `'x'` into the integration variable field.

***Note that when copying multi-line text from this pdf document and pasting it into the Numerical Integrator application, the pdf line endings may disappear, or may not be compatible. If the Numerical Integrator gives an error message, then it may be necessary to copy over individual lines, or else manually enter the expressions.***

The input fields should appear as shown below.

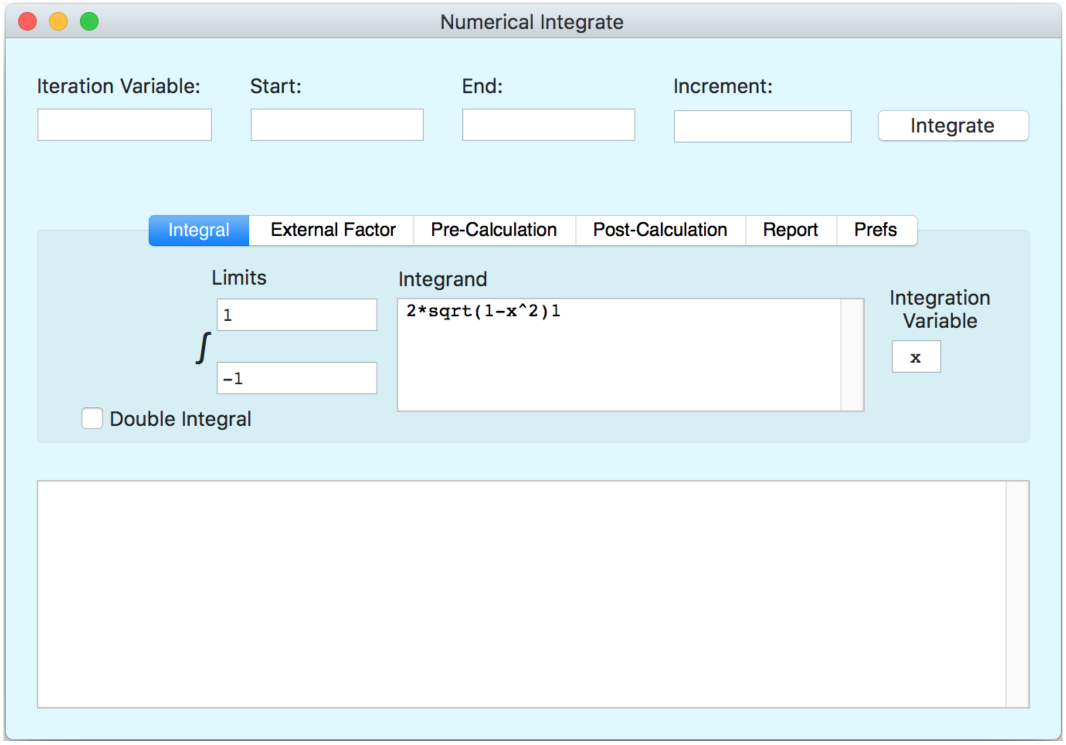


*Figure 2*

For the moment, we will leave the text fields in the upper part of the window with their default values which should be as shown above.

Now, click the **Integrate** button. The first run after starting the program will require a few seconds to initialize the set of internal calculation tables. The actual integration will then proceed quickly, and the result will be displayed in the output text field in the lower part of the window:

```
3.14159265358979323846264338 33
```

As can be seen, this is the expected value, equal to # and precise to 29 significant digits.

**Example 2 - Double Integral**

The next example shows how to evaluate double integrals. We will look at the formula for a hemispherical surface and integrate to get the volume under that surface. The volume of a sphere is $\frac{4}{3}\pi r^3$, and thus the volume of a hemisphere is half of that, or $\frac{2}{3}\pi r^3$. Again, we can use a radius of 1 and then include the factor $\frac{3}{2}$ so that the final result should again be $\pi$. The integral is:

$$\int_u^1 \int_{-\sqrt{1-y^2}}^{\sqrt{1-y^2}} \frac{3}{2}\sqrt{1-x^2-y^2}\; dx\, dy$$

The double integral checkbox must be checked. Then, the additional input fields for the outer integral will appear. On the far left are the additional limits, and on the far right is the additional variable of integration.

The integrand in computer notation is:

```
3/2*sqrt(1-x^2-y^2)
```

We enter the integrand, the limits and the variables of integration as shown below:

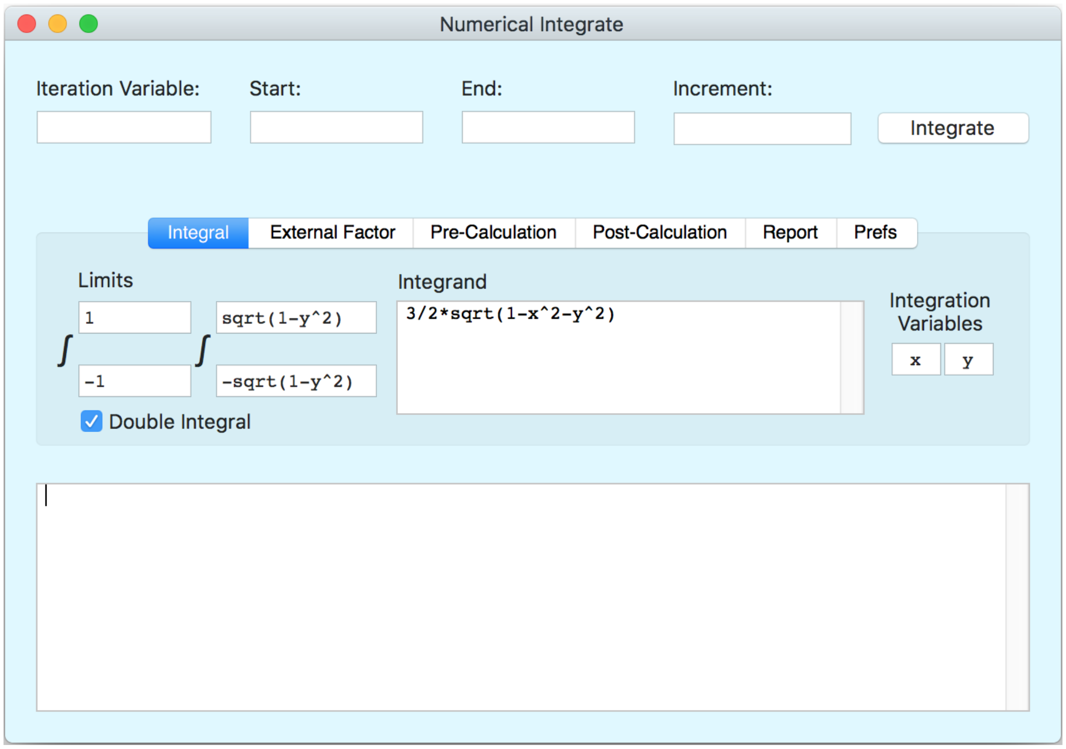


*Figure 3*

Then, clicking the Integrate button will result in the expected value of !, once again to the same precision as in the previous example.

```
3.14159265358979323846264338 33
```

As can be seen, the limits of integration do not have to be fixed constants. They can be expressions including variables and functions. Any expression that makes mathematical sense in the context of the integral is acceptable for the limits.

**External Factor Tab**

Normally, when evaluating the integral of the previous example, we would move the factor of $3/2$ outside of the integral, because it's a fixed constant. We can do this using the **External Factor** entry field. Delete `'3/2*'` from the **Integrand** field, then select the **External Factor** tab, and enter `'3/2'` in the entry field.

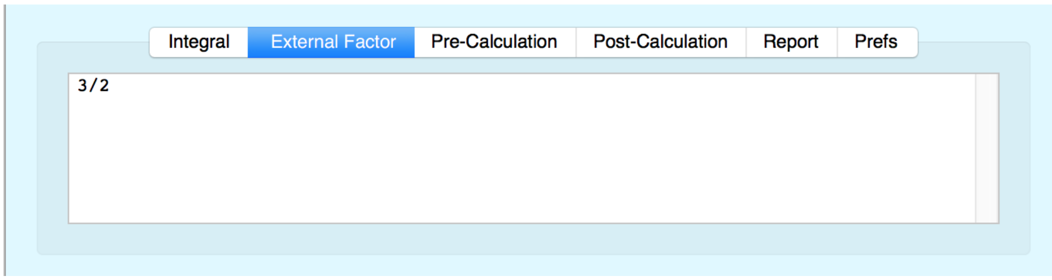


*Figure 4*

The External Factor field is not limited to numeric values, it can include expressions involving variables and functions. Clicking the Integrate button, will again return the value of $\pi$.

**Solving for multiple values of inputs**

In many cases we may wish to evaluate the integral for multiple different values of some variable. In this next example, this will be demonstrated. We will use a single integral but with a more elaborate external factor:

$$\frac{4}{3} \times \frac{4\pi\times10^{-7}}{(1-u)^3}\int_u^1 \left(1-\left(\frac{u}{k}\right)^3\right)(K(k)-E(k))\ dk$$

This is the formula for the inductance of a disk current sheet having an outer diameter of 1 meter, and an inner diameter of $u$ meters. The functions $\mathbf{K}(k)$ and $\mathbf{E}(k)$ are the complete elliptic integrals of the first and second kind to modulus $k$. We wish to generate a table of inductance values for values of the inner radius $u$ ranging from 0 to 1 in steps of 0.2.

The procedure is as follows:

- At the top right of the window, enter `'u'` as the Iteration Variable; enter `0` in the Start field, `1` in the End field, and `0.2` in the Increment field.
- Uncheck the Double Integral checkbox;
- In the Integrand entry field, enter: `(1-(u/k)^3)*(elipK(k)-elipE(k))`
- For the limits, enter `'u'` for the lower one and `1` for the upper one;
- Enter `'k'` as the variable of integration.
- Click the External Factor tab and enter: `4/3*4e-7*pi/(1-u)^2`

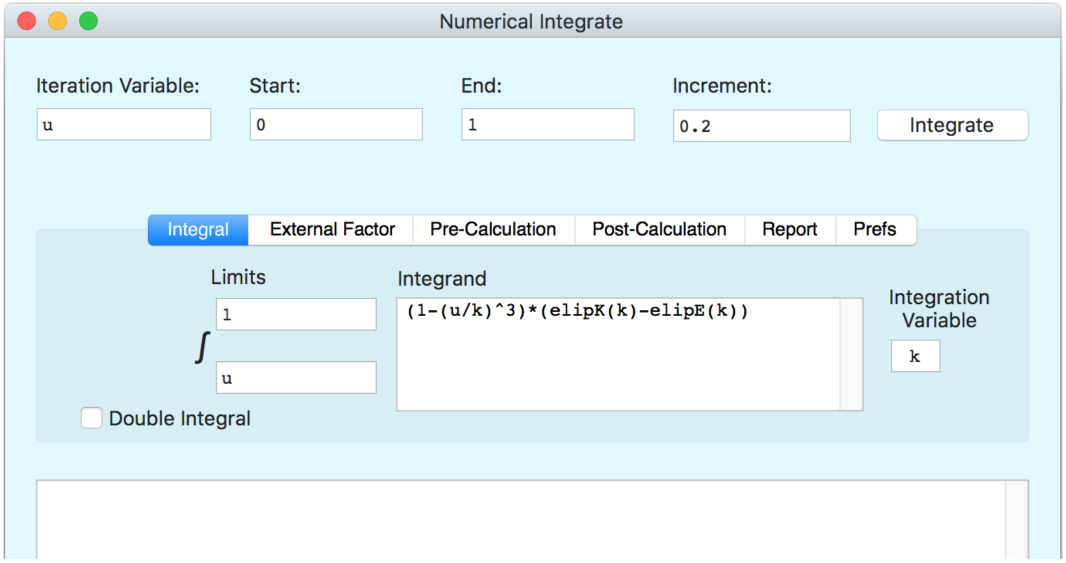


*Figure 5*

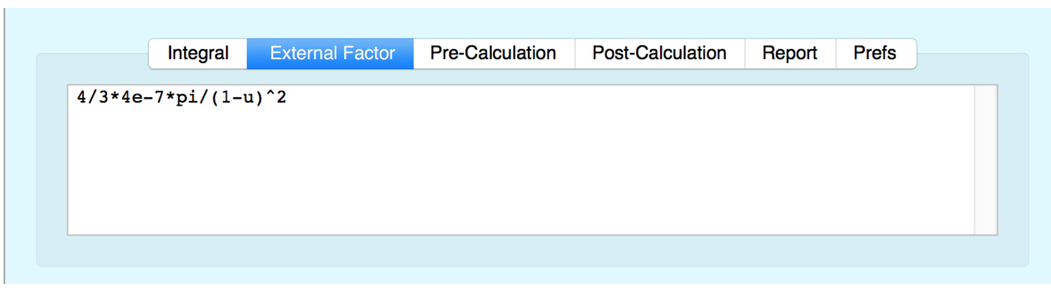


*Figure 6*

Click the **Integrate** button. The results should be as shown below. The value of the iterated variable is shown along with the value of the integral. The final value for ***u***=1 produces a **NaN** (not a number) result because of the division by zero in the external factor. Whenever an invalid operation occurs, the final result will be displayed as NaN.

```
Calculation Results
u, integral
0, 6.9695704256707441735800451994e-7
0.2, 1.0512457712388710360434144573e-6
0.4, 1.5645766022307272901997417184e-6
0.6, 2.3013638415869544312570619088e-6
0.8, 3.4915507837780520395982785045e-6
1.e0, NaN
```

Also note that the values entered into the **Start**, **End** and **Increment** fields are interpreted as floating point numbers to the same precision as specified for all other operations. Because some decimal values are represented as repeating binary values, it is sometimes possible that the value of the iterating variable may be very slightly higher than the end value on the last increment. This will prevent the final calculation from being performed. To avoid this, it is recommended that the **End** value be set slightly higher than the desired value (up to half of the increment is a safe value), to ensure that the final calculation is performed.

**Pre- and Post-Calculations**

The **Pre-Calculation** and **Post-Calculation** tabs will now be described.

It is often useful to perform additional calculations that relate to the evaluation of the integral, but are not part of the integral formula itself. These could be done using a separate application such as a spreadsheet, but it may be more advantageous to do it at the same time as the integral evaluation, and to the same precision that is available with this application. The Pre-Calculation tab provides an entry field where the user can enter one or more expressions to be evaluated, after the iteration variable's value is set, but before the integral is evaluated. Similarly, the Post- Calculation tab provides an entry field where the user can enter one or more expressions to be evaluated, after the integral is evaluated, and before the iteration variable is incremented for the next round. The results of these calculations can appear in the output text field. Also, the results of the Pre-Calculations can be used in the integral expression.

In the previous example, the integral was evaluated for a linearly spaced set of ***u*** values. Suppose we wish to plot the output data on a semi-log or log-log graph. It may be more useful to evaluate the integral for a series of ***u*** values that are spaced logarithmically. The iteration data entry fields do not offer this option, but we can convert that set of linearly spaced values to a set of logarithmically spaced ones. For this next example we will recalculate the integral of the previous example, but with a set of ***u*** values ranging from 0.001 to 1, with 5 values per decade. This is 3 decades with 5 values per decade, a total of 15 increments. The final value is 1 and its base 10 logarithm is 0. Therefore, we will use 0 as the end value, –15 as the start value, and 1 as the increment.

We will change the iteration variable name from ***u*** to ***i*** so that the set of ***i*** values will be linearly spaced and we will calculate the corresponding values of ***u*** in the **Pre-Calculation** tab. We will set the variable ***n*** = 5 which is the required number of ***u*** values per decade. Then the value of ***u*** is given by:

```
u=10^(i/n)
```

To complete this example, do the following:

- At the top right of the window, enter `'u'` as the Iteration Variable; enter `0` in the Start field, `1` in the End field, and `0.2` in the Increment field.
- Uncheck the Double Integral checkbox;
- In the Iteration Variable field at the top left of the window, change the iteration variable name from `u` to `i`;
- Set the Start value to `–15`, the End value to `0` and the Increment to `1`
- In the Pre-Calculation tab entry field, enter the following expressions, each one on a separate line:

```
n=5
u=10^(i/n)
```

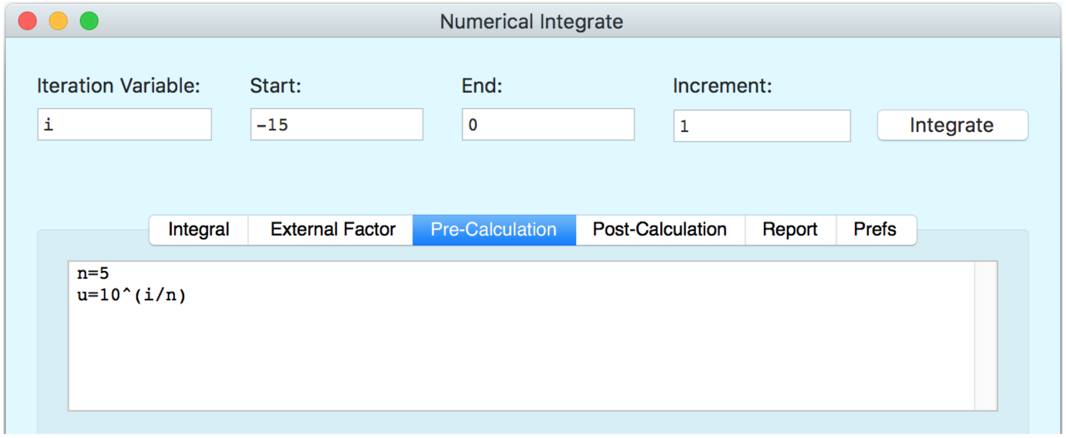


*Figure 7*

Now, click the **Integrate** button. The result will be as shown below.

```
Calculation Results
i, integral
-1.5e1, 6.9835304029637676325808333799e-7
-1.4e1, 6.99171473152907649239770645222e-7
-1.3e1, 7.00471493318034075648519151355e-7
-1.2e1, 7.02539109075200956934402528282e-7
-1.1e1, 7.05834063992084284162326088744e-7
-1.e1, 7.11100983993621963975378445558e-7
-9.e0, 7.19559467387742161257124577742e-7
-8.e0, 7.33239457480958772305967720073e-7
-7.e0, 7.55596222194639460018424422243e-7
-6.e0, 7.92690823483693013049717268885e-7
-5.e0, 8.55580786572349527775465862237e-7
-4.e0, 9.65496216604385491019776884418e-7
-3.e0, 1.16618622677579758931193245888e-6
-2.e0, 1.55883722771311492382331761669e-6
-1.e0, 2.4452054345452673145288834966e-6
0, NaN
```

Note that only the values of ***i*** and the integral are displayed. It would be useful to display the value of ***u*** as well. The next section explains how to choose which variables to display.

**Report Tab**

The purpose of the **Report** Tab entry field is to list the variables whose values are to be displayed in the output. It is simply a list of variable names separated with commas. If this field is left blank, then the default output will be the iteration variable value and the integral value. If the field is not blank, then the variables listed will be those shown in the output, in the order given. The variable name '`integral`' is predefined as the value of the evaluated integral (multiplied by the external factor). Any other variables that have been defined by the user can also be listed here.

In the **Report** tab entry field, enter: `i,u,integral`.

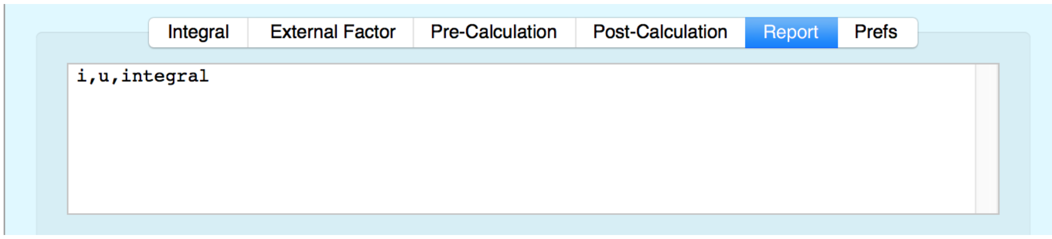


*Figure 8*

Then click the **Integrate** button. The result will be:

```
Calculation Results
i, u, integral
-1.5e1, 0.001, 6.98353040296376763258083337999e-7
-1.4e1, 0.00158489319246111348520210137334, 6.99171473152907649239770645222e-7
-1.3e1, 0.00251188643150958011108503206678, 7.00471493318034075648519151355e-7
-1.2e1, 0.00398107170553497250770252305509, 7.02539109075200956934402528288e-7
-1.1e1, 0.00630957344480193249434360136662, 7.05834063992084284162326088744e-7
-1.e1, 0.01, 7.11100983993621963975378445558e-7
-9.e0, 0.0158489319246111348520210137334, 7.19559467387742161257124577742e-7
-8.e0, 0.0251188643150958011108503206678, 7.33239457480958772305967720733e-7
-7.e0, 0.0398107170553497250770252305509, 7.55596222194639460018424422433e-7
-6.e0, 0.0630957344480193249434360136662, 7.92690823483693013049717268855e-7
-5.e0, 0.1, 8.55580786572349527775465862377e-7
-4.e0, 0.158489319246111348520210137334, 9.65496216604385491019776884188e-7
-3.e0, 0.251188643150958011108503206678, 1.16618622677579758931193245888e-6
-2.e0, 0.398107170553497250770252305509, 1.55883722771311492382331761699e-6
-1.e0, 0.630957344480193249434360136662, 2.44520543454526731452888349966e-6
0, 1.e0, NaN
```

In addition to the predefined variable '`integral`', there are several other predefined variables that provide information about the result of the integration:

`erri` - The estimated absolute error in the calculation of the inner integral
`erro` - The estimated absolute error in the calculation of the outer integral
`lci` - Inner Level Count: The number of times the inner integral is split into sub-partitions
`lco` - Outer Level Count: The number of times the outer integral is split into sub-partitions
`fci` - Total number of times the integrand is evaluated.
`fco` - Total number of times the outer integral routine calls the inner integral routine.

In the case of a single integral, the values of `erri`, `lci` and `fci` will apply. The variables `erro`, `lco` and `fco` will be zero.

If we now include `erri` in the list of report variables in the current example, and run the integration again, the following is the result:

```
Calculation Results
i, u, integral, erri
-15, 0.001, 6.98353040296376763258083337996e-7, 3.7386108977061956247830171782e-26
-14, 0.0015848931924611134852021013734, 6.9917147315290764923977064522e-7, 2.0010487680663636485524895862e-26
-13, 0.0025118864315095801110850320678, 7.0047149331803407564851915135e-7, 5.5801964979568620935549641439e-27
-12, 0.0039810717055349725077025230509, 7.0253910907520095693440252828e-7, 1.2064797424512359628271939357e-27
-11, 0.0063095734448019324943436013662, 7.0583406399208428416232608874e-7, 2.7552791679556965278492226617e-28
-10, 0.01, 7.1110098399362196397537844558e-7, 5.5957608250161133647610201637e-29
-9, 0.015848931924611134852021013734, 7.1955946738774216125712457742e-7, 5.8984596558481714654303838 58e-30
-8, 0.025118864315095801110850320678, 7.3323945748095877230596772073e-7, 3.4439857600045571308000402708e-31
-7, 0.039810717055349725077025230509, 7.5559622219463946001842442243e-7, 3.4331002519873575291440778244e-31
-6, 0.063095734448019324943436013662, 7.9269082348369301304971726885e-7, 3.4325324442506724975348681478e-31
-5, 0.1, 8.5558078657234952777546586237e-7, 3.2327063207172355996678022476e-31
-4, 0.15848931924611134852021013734, 9.6549621660438549101977688418e-7, 1.8970640207673264597598731994e-40
-3, 0.25118864315095801110850320678, 1.1661862267757975893119324588e-6, 2.0193020142933567888212573078e-44
-2, 0.39810717055349725077025230509, 1.5588372277131149238233176169e-6, 6.5474906515169763130362211333e-47
-1, 0.63095734448019324943436013662, 2.4452054345452673145288883496e-6, 9.5658557986349394713292927784e-52
0, 1, NaN, 0
```

## Prefs Tab - Integration Precision Parameters

The items in the Prefs tab are beyond the scope of a Quick Start Guide, but are mentioned here for the sake of completeness. They are used for adjusting the precision or number of significant decimal places of the calculation. These items are;

- Truncation Limit
- Partition Levels
- Precision (digits)
- TargetError

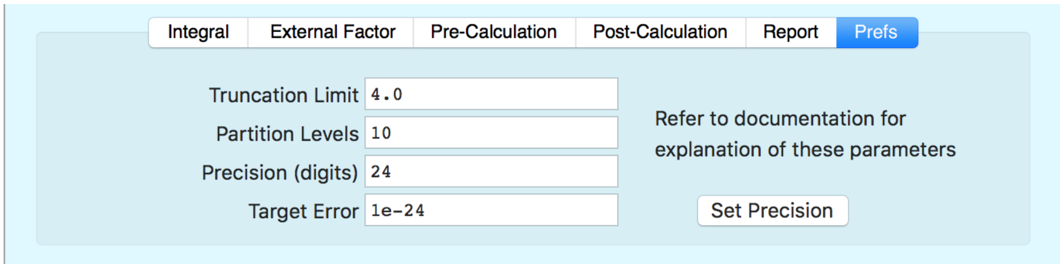


*Figure 9*

***The following explanation is cursory, and a future revision of this document will contain a more detailed discussion.***

Starting at the bottom, the **Target Error** value is the desired maximum absolute error of the integration. A smaller value will produce a more precise result. A larger value will produce faster but less precise results. The target error pertains to the precision of the Double Exponential integration technique.

While the **Target Error** value relates to the integration algorithm, the **Precision** value relates to the extended precision math package, and is the number of decimal digits of precision that are to be used for the calculations. There is no limit to the precision except for the available computer memory. Typically, the precision value will be the same value as the exponent value in the Target Error value (24 in Figure 9).

The **Partition Levels** is the number of times that the number of integration intervals may be doubled. This parameter relates to the integration algorithm and affects precision. If the integration fails to produce a result to the precision entered in the Precision field, then the integration intervals value should be increase. However, be warned that for every increase in this value by 1, it will double the time taken to regenerate the integration tables.

The **Truncation Limit** again affects the precision of the integration algorithm. The recommended values for this parameter depend on the number of digits of precision and are given in the table below.

| Truncation Limit | Resulting Significant Digits |
|---|---|
| 1 | 1 |
| 2 | 4 |
| 3 | 12 |
| 3.25 | 16 |
| 4 | 35 |
| 4.5 | 59 |
| 5 | 99 |

This concludes the **Quick Start** part of the documentation.

## Reference Information

### About the Algorithm

This application uses the Double Exponential numerical integration method as described by Takahashi and Mori [1]. The method takes advantage of the fact that for many bell shaped functions, the trapezoidal rule gives extremely fast (exponential) convergence. The Double Exponential numerical integration method transforms the input function into such a bell shaped function, and then performs the numerical integration using the trapezoidal rule.

The Double Exponential numerical integration method also has the advantage that it can integrate expressions which have singularities occurring at either or both limits of integration. To implement this method, the original integral over a finite range is transformed into an integral over the interval $(-\infty, \infty)$, which is then evaluated using the trapezoidal rule.

***The current version of the Reference Information section ends here. In a future revision, additional details will be added.***

## References


1. Takahasi, H & Mori, M.; *Double Exponential Formulas for Numerical Integration*. Publ. RIMS, Kyoto University 9 (1974) 721–741.

2. Delaney, Robert; *Bob Delaney's Science Software*; Web page:
   http://delaneyrm.com/
   (Downloaded on 2021-05-14)
   Note that a fork of this extended precision math package is now hosted by
   Björn Eiríksson:
   https://einhugur.com/Html/opensource.html
   (Downloaded on 2025-01-27)

3. Cook, John D.; *Double Exponential Integration*; Web page:
   https://www.johndcook.com/blog/double_exponential_integration/
   *Fast Numerical Integration;* Web page:
   https://www.codeproject.com/Articles/31550/Fast-Numerical-Integration
   (Downloaded on 2017-06-09)

# Appendix A

## List of Mathematical Operators and Functions

### Arithmetic Operators

+ - Addition
- - Subtraction
* - Multiplication
/ - Division
\ - Integer Division (floating point result)
mod - Remainder from integer division (floating point result)
^ - Exponentiation
mn - minimum
mx - maximum

### Comparison and Logical Operators

These return 1 if the result is true, and 0 if false. Operands equal to zero are considered to be false, and non-zero operands are considered to be true. Note that AND,
OR, NOT and XOR are not bitwise.

<
<=
<>
== - Is equal to (Do not use a single = symbol)
>

>=
AND
NOT
OR
XOR

### Constant Functions

These are functions which return a constant value.

Pi - 3.14...
Eulergamma - 0.57721... Euler-Mascheroni Constant

### Functions in Alphabetical order

The meaning of most of these functions are self-evident. Explanations are given for nonstandard ones.

```
abs(x)
acos(x)
acosh(x)
asin(x)
asinh(x)
atan(x)
atan2(x,y)
atanh(x)
ceil(x)
cos(x)
cosh(x)
ElipK(k) - Complete Elliptic Integral of First Kind K with modulus k
ElipE(k) - Complete Elliptic Integral of Second Kind E with modulus k
ElipKE(k) - K-E (calculated directly without subtraction)

exp(x)
factorial(x) - Depending on the value of x, this is either x! or Gamma(x+1)
floor(x)
gamma(x) - Gamma function
```

— !16 —

```
if(a,b,c) - If 'a' is true, returns b; otherwise returns c
int(x) - argument truncated (not rounded) to integer
lnfactorial - Ln(Gamma(x+1))
log(x) - natural logarithm

max(x,y) - maximum of x and y (see also mx operator)
min(x,y) - minimum of x and y (see also mn operator)
round(x) - Round to nearest integer value
sin(x)

sinh(x)
sqrt(x)
tan(x)
tanh(x)
```

(Note that functions ceil, floor, int, and round, return a floating point value)

## Special Functions

```
beta(a,b) - Beta Function: = Integral(0 to 1) t^(a-1) (1-t)^(b-1) dt; a > 0 , b > 0

ibeta(a,b,x) - Incomplete Beta Function: = Integral(0 to x) t^(a-1) (1-t)^(b-1) dt;
a > 0 , b > 0 , 0 <= x <= 1

erf(x) - Error function: = (2 / sqrt(Pi)) Integral(0 to x) e^(-t^2) dt

erfc(x) - Error function: = 1-erf() = (2 / sqrt(Pi)) Integral(x to inf) e^(-t^2) dt

expint(n,x) - Exponential Integral: = Integral(1 to inf) e^(-x*t)/t^n; n >= 0 , x
>= 0 but cannot have x = 0 when n <= 1

fresnelC(x) - Fresnel's Integral: = Integral(0 to x) cos(Pi*t^2/2) dt
fresnelS(x) - Fresnel's Integral: = Integral(0 to x) sin(Pi*t^2/2) dt
besselI(n,x) - order n, argument x
besselJ(n,x)

besselK(n,x)
besselY(n,x)
besselJzero(n,x) - x is estimate of zero location, returns accurate location
besselYzero(n,x) - x is estimate of zero location, returns accurate location
sphBesselJ(n,x) - Spherical Bessel J
sphBesselY(n,x) - Spherical Bessel Y
kummerm(a,b,x) - Kummer's Confluent Hypergeometric Function M
```

--- End of Document ---