integral

Numerical integration

Syntax

```
q = integral(fun,xmin,xmax)
q = integral(fun,xmin,xmax,Name,Value)
```

Description

`q = integral(fun,xmin,xmax)` numerically integrates function `fun` from `xmin` to `xmax` using global adaptive quadrature and default error tolerances. example

`q = integral(fun,xmin,xmax,Name,Value)` specifies additional options with one or more `Name,Value` pair arguments. For example, specify `'WayPoints'` followed by a vector of real or complex numbers to indicate specific points for the integrator to use. example

Examples

collapse all

Improper Integral

Create the function $f(x) = e^{-x^2}(\ln x)^2$.

Open in MATLAB Online

Copy Command

```
fun = @(x) exp(-x.^2).*log(x).^2;
```

Evaluate the integral from `x=0` to `x=Inf`.

```
q = integral(fun,0,Inf)
```

```
q = 1.9475
```

Parameterized Function

Create the function $f(x) = 1/(x^3 - 2x - c)$ with one parameter, c.

Open in MATLAB Online

Copy Command

```
fun = @(x,c) 1./(x.^3-2*x-c);
```

Evaluate the integral from `x=0` to `x=2` at `c=5`.

```
q = integral(@(x) fun(x,5),0,2)
```

```
q = -0.4605
```

See Parameterizing Functions for more information on this technique.

Singularity at Lower Limit

Create the function $f(x) = \ln(x)$.

Open in MATLAB Online

Copy Command

```
fun = @(x)log(x);
```

Evaluate the integral from `x=0` to `x=1` with the default error tolerances.

```
format long
q1 = integral(fun,0,1)
```

```
q1 =
  -1.000000010959678
```

Evaluate the integral again, this time with 12 decimal places of accuracy. Set `RelTol` to zero so that `integral` only attempts to satisfy the absolute error tolerance.

```
q2 = integral(fun,0,1,'RelTol',0,'AbsTol',1e-12)
```

```
q2 =
  -1.000000000000010
```

Complex Contour Integration Using Waypoints

Create the function $f(z) = 1/(2z - 1)$.

Open in MATLAB Online

Copy Command

```
fun = @(z) 1./(2*z-1);
```

Integrate in the complex plane over the triangular path from `0` to `1+1i` to `1-1i` to `0` by specifying waypoints.

```
q = integral(fun,0,0,'Waypoints',[1+1i,1-1i])
```

```
q = 0.0000 - 3.1416i
```

Vector-Valued Function

Create the vector-valued function $f(x) = [\sin x, \sin 2x, \sin 3x, \sin 4x, \sin 5x]$ and integrate from `x=0` to `x=1`. Specify `'ArrayValued',true` to evaluate the integral of an array-valued or vector-valued function.

Open in MATLAB Online

Copy Command

```
fun = @(x)sin((1:5)*x);
q = integral(fun,0,1,'ArrayValued',true)
```

```
q = 1×5

    0.4597    0.7081    0.6633    0.4134    0.1433
```

Improper Integral of Oscillatory Function

Create the function $f(x) = x^5 e^{-x} \sin x$.

Open in MATLAB Online

Copy Command

```
fun = @(x)x.^5.*exp(-x).*sin(x);
```

Evaluate the integral from `x=0` to `x=Inf`, adjusting the absolute and relative tolerances.

```
format long
q = integral(fun,0,Inf,'RelTol',1e-8,'AbsTol',1e-13)
```

```
q = 
 -14.999999999998360
```

Input Arguments

collapse all

`fun` — Integrand
function handle

Integrand, specified as a function handle, which defines the function to be integrated from `xmin` to `xmax`.

For scalar-valued problems, the function `y = fun(x)` must accept a vector argument, `x`, and return a vector result, `y`. This generally means that `fun` must use array operators instead of matrix operators. For example, use `.*` (`times`) rather than `*` (`mtimes`). If you set the `'ArrayValued'` option to `true`, then `fun` must accept a scalar and return an array of fixed size.

`xmin` — Lower limit of x
real number | complex number

Lower limit of x, specified as a real (finite or infinite) scalar value or a complex (finite) scalar value. If either `xmin` or `xmax` are complex, then `integral` approximates the path integral from `xmin` to `xmax` over a straight line path.

Data Types: `double` | `single`
Complex Number Support: Yes

`xmax` — Upper limit of x
real number | complex number

Upper limit of x, specified as a real number (finite or infinite) or a complex number (finite). If either `xmin` or `xmax` are complex, `integral` approximates the path integral from `xmin` to `xmax` over a straight line path.

Data Types: `double` | `single`
Complex Number Support: Yes

Name-Value Arguments

Specify optional pairs of arguments as `Name1=Value1,...,NameN=ValueN`, where `Name` is the argument name and `Value` is the corresponding value. Name-value arguments must appear after other arguments, but the order of the pairs does not matter.

Before R2021a, use commas to separate each name and value, and enclose `Name` in quotes.

Example: `integral(fun,a,b,'AbsTol',1e-12)` sets the absolute error tolerance to approximately 12 decimal places of accuracy.

`AbsTol` — Absolute error tolerance
`1e-10` (default) | nonnegative real number

Absolute error tolerance, specified as the comma-separated pair consisting of `'AbsTol'` and a nonnegative real number. `integral` uses the absolute error tolerance to limit an estimate of the absolute error, $|q - Q|$, where q is the computed value of the integral and Q is the (unknown) exact value. `integral` might provide more decimal places of precision if you decrease the absolute error tolerance.

> **Note**
>
> `AbsTol` and `RelTol` work together. `integral` might satisfy the absolute error tolerance or the relative error tolerance, but not necessarily both. For more information on using these tolerances, see the Tips section.

Example: `integral(fun,a,b,'AbsTol',1e-12)` sets the absolute error tolerance to approximately 12 decimal places of accuracy.

Data Types: `single` | `double`

`RelTol` — Relative error tolerance
`1e-6` (default) | nonnegative real number

Relative error tolerance, specified as the comma-separated pair consisting of `'RelTol'` and a nonnegative real number. `integral` uses the relative error tolerance to limit an estimate of the relative error, $|q - Q|/|Q|$, where q is the computed value of the integral and Q is the (unknown) exact value. `integral` might provide more significant digits of precision if you decrease the relative error tolerance.

> **Note**
>
> `RelTol` and `AbsTol` work together. `integral` might satisfy the relative error tolerance or the absolute error tolerance, but not necessarily both. For more information on using these tolerances, see the Tips section.

Example: `integral(fun,a,b,'RelTol',1e-9)` sets the relative error tolerance to approximately 9 significant digits.

Data Types: `single` | `double`

`ArrayValued` — Array-valued function flag
`false` or `0` (default) | `true` or `1`

Array-valued function flag, specified as the comma-separated pair consisting of `'ArrayValued'` and a numeric or logical `1` (`true`) or `0` (`false`). Set this flag to `true` or `1` to indicate that `fun` is a function that accepts a scalar input and returns a vector, matrix, or N-D array output.

The default value of `false` indicates that `fun` is a function that accepts a vector input and returns a vector output.

Example: `integral(fun,a,b,'ArrayValued',true)` indicates that the integrand is an array-valued function.

`Waypoints` — Integration waypoints
vector

Integration waypoints, specified as the comma-separated pair consisting of `'Waypoints'` and a vector of real or complex numbers. Use waypoints to indicate points in the integration interval that you would like the integrator to use in the initial mesh:

- Add more evaluation points near interesting features of the function, such as a local extrema.
- Integrate efficiently across discontinuities of the integrand by specifying the locations of the discontinuities.
- Perform complex contour integrations by specifying complex numbers as waypoints. If `xmin`, `xmax`, or any entry of the waypoints vector is complex, then the integration is performed over a sequence of straight line paths in the complex plane. In this case, all of the integration limits and waypoints must be finite.

Do not use waypoints to specify singularities. Instead, split the interval and add the results of separate integrations with the singularities at the endpoints.

Example: `integral(fun,a,b,'Waypoints',[1+1i,1-1i])` specifies two complex waypoints along the interval of integration.

Data Types: `single` | `double`
Complex Number Support: Yes

Tips

- The `integral` function attempts to satisfy:

```
abs(q - Q) <= max(AbsTol,RelTol*abs(q))
```

where `q` is the computed value of the integral and `Q` is the (unknown) exact value. The absolute and relative tolerances provide a way of trading off accuracy and computation time. Usually, the relative tolerance determines the accuracy of the integration. However if `abs(q)` is sufficiently small, the absolute tolerance determines the accuracy of the integration. You should generally specify both absolute and relative tolerances together.

- If you are specifying single-precision limits of integration, or if `fun` returns single-precision results, you might need to specify larger absolute and relative error tolerances.

References

[1] L.F. Shampine "*Vectorized Adaptive Quadrature in MATLAB®*," *Journal of Computational and Applied Mathematics*, 211, 2008, pp.131–140.

Extended Capabilities

> **C/C++ Code Generation**
Generate C and C++ code using MATLAB® Coder™.

> **Thread-Based Environment**
Run code in the background using MATLAB® `backgroundPool` or accelerate code with Parallel Computing Toolbox™ `ThreadPool`.

Version History

Introduced in R2012a

See Also

`integral2` | `integral3` | `trapz`

Topics

Integration of Numeric Data
Integration to Find Arc Length
Complex Line Integrals
Create Function Handle
Parameterizing Functions