
Integration

Table of Contents

Indefinite Integrals	1
Where's the +C?	1
Definite Integrals	1
Approximating Integrals	1
Using quad	2

Integration can be indefinite or definite. Matlab can do both. If the definite integration is too tricky or impossible, Matlab can approximate.

Indefinite Integrals

Suppose you'd like to integrate the function $7/(1+x^2)$. Hmm... that reminds me of something I've seen before, but I can't recall the formula... Something about an inverse trig function... Ah, let's just ask Matlab:

```
int(7/(1 + x^2))
```

```
ans =
```

```
7*atan(x)
```

Oh, yeah -- that's it! Matlab uses `atan` for arctangent which is the same as inverse tangent.

Where's the +C?

Matlab doesn't put the +C. You still need to on your exams.

Definite Integrals

Suppose we'd like to compute the area under the curve $f(x) = x^3 + \ln x$ over the interval where x is between 5 and 10. Just pass those end-points to the `int` command as two additional parameters:

```
int(x^3+log(x),5,10)
```

```
ans =
```

```
5*log(20) + 9355/4
```

Approximating Integrals

We've seen how the `int` command can be used to find definite integrals. You should know that `int` does its job using *symbolic* integration techniques, similar to the ones you would use if you were integrating

something with paper and pencil in a first semester calculus course. Unfortunately, there are many functions that are difficult or impossible to integrate this way. For example, I tried the following:

```
int(sqrt(sin(x).*5), 1, 2)

ans =

-2*5^(1/2)*(ellipticE(pi/4 - 1, 2) - ellipticE(pi/4 - 1/2, 2))
```

That answer is not really what I was after -- I just wanted a number! What in the world is `ellipticE` anyway?!

Using quad

To handle this definite integral, it might be better to ask Matlab to use a *numerical* method to compute the answer. There are a few different commands that are built-in to Matlab for handling this task. One of them is called `quad` which is short for quadrature -- a word that actually means *approximate the definite integral using a numerical method*. (For now don't worry about the actual method used, just remember that it's like the ones you know - Simpson's Rule and so on.) Here is `quad` in action. I'll change the format for lots of detail, too:

```
format long
quad(@(x)sqrt(sin(x).*5), 1, 2)

ans =

2.186306000969553
```

There are two things that should concern you here. First that `@(x)` symbol. We'll explain this in more detail later but basically the first argument of `quad` must be a something called a *function handle* and the way to make a function into a function handle is to put an `@(x)` (or whatever variable you're using) in front.

The second issue is that `.*` that replaced the `*` by itself. The reason for this is that `quad` actually does its job in the background using vectors. We don't particularly need to care what Matlab is doing in the background with the `quad` function but we need to make sure we give it the correct format.

So basically just remember to put `@(x)` in front and in place of `*`, `/` and `^` use `.*`, `./` and `.^` and you'll be fine.

Published with MATLAB® 8.0