Solving Equations Numerically

Table of Contents

Next we'll talk a little about what to do when symbolic approaches don't work.

What are Numerical Methods?

Some equations can't be solved symbolically. For these, we can still ask Matlab to try to find very close approximations using numerical methods. Later, we'll show some more advanced examples, but for now, let's try an easy one. Suppose we want to find the solution to the equation `log(x)+x+1=0`. (Remember that Matlab uses `log` for natural log!) We can try to solve this symbolically as shown below:

```
syms x
solve(log(x) + x + 1)
```

```
ans =

lambertw(0, exp(-1))
```

Although this answer is ugly, it is 100% exact. [By the way, `exp(1)` is the constant `e=2.7183...` The problem with this answer is that unless you are already familiar with the Lambert W (Omega) function, this ugly formula will not be very useful for you. Even if you *do* know what Lambert's W function is, you still have to do more work to figure out what actual numerical value the above formula represents. Sometimes you just want to see a number, even if it isn't exactly precise!

Solving an Equation Numerically using `fzero`

Instead of having Matlab solve the equation symbolically, let's try to do it numerically. We'll employ a command `fzero` which will use some sophisticated numerical techniques to find the zeros of any equation. To use the `fzero` command we have to first come up with an `estimate` of the correct answer. The `fzero` command will take our estimate and repeatedly apply an algorithm that will obtain a better and better approximation. Let's say our starting estimate is the value 2. We would call `fzero` like this:

```
syms x
fzero('log(x) + x + 1', 2)
```

```
ans =

    0.2785
```

Choosing Well

It's important to give `fzero` a good initial guess. For example the equation `sin(1/x)=0` has many solutions but only one of them near `0.3`. If we do

```
fzero('sin(1/x)',0.1)
```

```
ans =

    0.1061
```

we don't get the one we want because `0.1` is not near that one. Instead we have to start close to the one we want.

```
fzero('sin(1/x)',0.3)
```

```
ans =

    0.3183
```

Why the Single Quotes?

A small note that `fzero` is a particularly picky Matlab function. The expression you give it *must* be in single quotes or it must be a function handle. We'll talk about function handles later so for now just remember that to use `fzero` you must put the expression in single quotes.

Is that it?

We'll talk more about `fzero` later, specifically about making good first guesses and function handles, just for now notice that it exists. Matlab has other methods of approximating solutions too, but this is the big one.

Published with MATLAB® 8.0