
Table of Contents

Solving Equations Numerically	1
Solving an Equation Numerically using <code>fzero</code>	1
Choosing Well	1
Using <code>fzero</code> with Symbolic Functions	2

Solving Equations Numerically

What happens when symbolic approaches don't work? For example consider the fairly simple 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==0)

ans =

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

Although this answer is ugly and almost certainly nonsensical, it is 100% exact. [By the way, `exp(1)` is the constant $e = 2.7183 \dots$. 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`

Matlab has a collection of tools for finding approximate solutions but we'll focus on just one, that's the `fzero` command. The `fzero` command takes an initial guess that we provide **which is close to the actual solution** and it repeatedly applies an algorithm that will obtain a better and better approximation until successive approximations are within a certain very small built-in tolerance of one another. Let's say our starting estimate is the value 2. We could call `fzero` like this:

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

ans =

0.2785
```

What this has done is find a root which is close to our initial guess of 2.

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 the one we want. Instead we have to start close to the one we want.

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

```
ans =
```

```
0.3183
```

Using `fzero` with Symbolic Functions

If you try the following you'll get an error:

```
syms f(x);  
f(x) = sin(1/x);  
fzero(f(x),0.3)
```

```
Error using fzero (line 169)  
If FUN is a MATLAB object, it must have an feval method.
```

```
Error in Ch08_SolvingNumerically (line 61)  
fzero(f(x),0.3)
```

The reason for this is that the `fzero` method can't handle symbolic functions. Don't ask me why, personally I consider it a flaw in Matlab. In any case in general `fzero` can only handle symbolic expressions contained in single quotes or function handles, which were briefly mentioned before and will be revisited shortly.

Published with MATLAB® 8.0