Chapter 07 - Solving Numerically

Table of Contents

Introduction 1
Solving Numerically using `vpasolve` 1

Introduction

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 =

wrightOmega(-1)
```

Although this answer is ugly and almost certainly nonsensical, it is 100% exact. The problem with this answer is that unless you are already familiar with the `wrightOmega` function, this ugly formula will not be very useful for you. Maybe you just want an approximate answer?

Solving Numerically using `vpasolve`

Matlab has a collection of tools for finding approximate solutions but we'll focus on just one, that's the `vpasolve` command. The `vpa` in `vpasolve` stands for **variable precision arithmetic**.

The `vpasolve` is powerful but not all-powerful. If we give it a polynomial equation it will find all solutions:

```
syms x
vpasolve(4*x^4+3*x^3+2*x^2==-x-5)
```

```
ans =

 - 0.880113771260681698178751904578350 +
0.763315833877154525129784681022630i
 - 0.880113771260681698178751904578350 -
0.763315833877154525129784681022630i
  0.505113771260681698178751904578350 +
0.815989650689463128532270678906560i
  0.505113771260681698178751904578350 -
0.815989650689463128532270678906560i
```

If we give it a non-polynomial solution it will find just one solution.

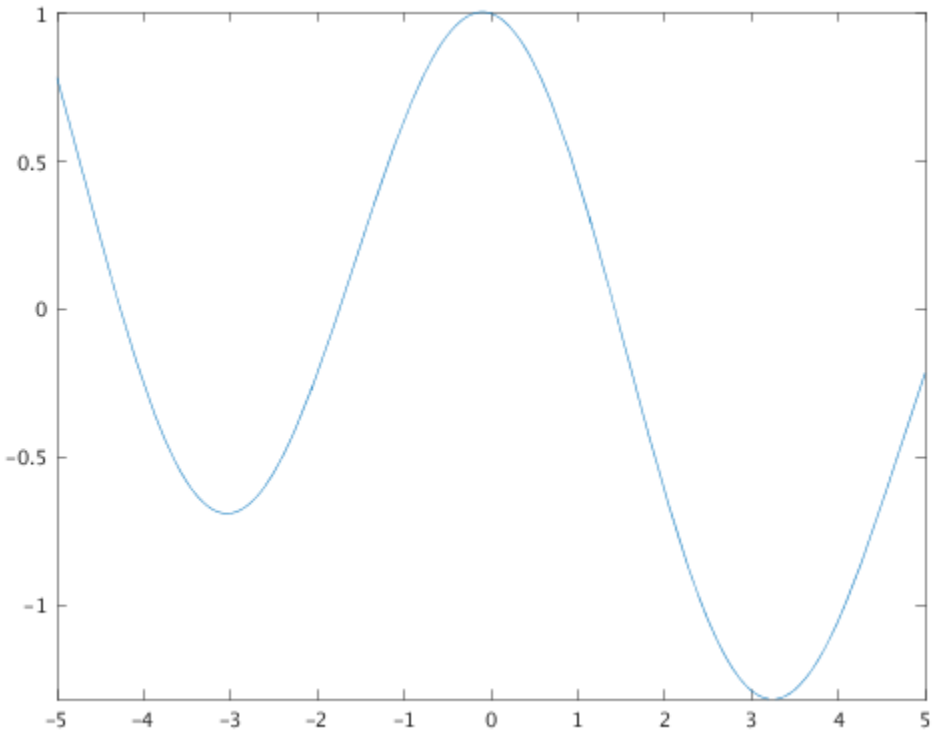
```
vpasolve(cos(x)-0.1*x==0)
```

```
ans =

5.267116434076329401608733749356 7
```

However this is not the only solution. If you graph the function `cos(x)-0.1*x` using Matlab's `fplot` command:

```
fplot(cos(x)-0.1*x)
```



You'll see many solutions. If this is your first time seeing a Matlab plot, notice that the x-axis (where `cos(x)-0.1*x==0` runs across the (almost vertical) middle of the rectangle. Check the y-axis on the left.

Now you can see that Matlab found the solution which lies off the right of the picture. But there's another solution near `x=1`. Can we find that one? Sure, we just tell Matlab to look close to that point by adding the varable and the starting point at the end of the command:

```
vpasolve(cos(x)-0.1*x==0,x,1)
```

```
ans =
```

```
1.427551778764594120806487988040 7
```

Generally `fplot` and `vpasolve` will be used hand-in-hand to plot the function, see approximately where the solution is, and then use that guess as a starting value.

Just to close, here's our original problem, finding one solution:

```
vpasolve(log(x) + x + 1==0)
```

```
ans =

0.27846454276107379510935873902298
```

Published with MATLAB® R2016b