Function Handles

Table of Contents

Function Handles ................................................................. 1
Differentiation and Integration .................................................. 2
Numerical Integration of Function Handles .................................. 3
The \texttt{matlabFunction} Command ........................................ 3
Plotting Function Handles ....................................................... 4
More Complicated Calculations - IMPORTANT! ............................. 4

So far we have seen functions defined symbolically. Matlab also has functions which are defined as function handles. These are used for certain Matlab commands like \texttt{quad} and \texttt{fzero} and are also used when we need to pass functions to function m-files as arguments as we will do in the last chapter.

Function Handles

Function handles are defined using the @ operator. Consider the following example:

\begin{verbatim}
f = @(x) x^5-3*x
\end{verbatim}

\begin{verbatim}
f = @(x)x^5-3*x
\end{verbatim}

This tells Matlab to create a function for which \( x \) is the variable (this is what the \( @ (x) \) does) and for which the rule is \( x^2-x \). Here are some things we can do with function handles. We can plug things in:

\begin{verbatim}
f(3)
\end{verbatim}

\begin{verbatim}
ans =
234
\end{verbatim}

We can use \texttt{fzero} to approximate a root:

\begin{verbatim}
\texttt{fzero}(f,2)
\end{verbatim}

\begin{verbatim}
\texttt{ans} =
1.3161
\end{verbatim}
Differentiation and Integration

Differentiating is a bit obscure. We cannot simply do `diff(f)` because `diff` doesn't work as-is on function handles. However if we define x as symbolic then Matlab will accept `diff`. What's happening here is that f is a function handle but f(x) is symbolic because symbolic x is plugged in:

```matlab
clear all;
f = @(x) x^5-3*x;
syms x;
diff(f(x))
```

```
ans = 
5*x^4 - 3
```

and

```matlab
diff(f(x),2)
```

```
ans = 
20*x^3
```

Also, strangely, provided x is symbolically defined then the following will work:

```matlab
diff(f,x,2)
```

```
ans = 
20*x^3
```

Integration works the same way in that `int(f)` and `int(f,1,2)` will not work but provided x is symbolic:

```matlab
clear all;
f = @(x) x^5-3*x;
syms x;
int(f(x))
```

```
ans = 
(x^2*(x^4 - 9))/6
```

and

```matlab
int(f(x),1,2)
```
Function Handles

ans =
6

will work fine, as will:
int(f,x,1,2)

ans =
6

Numerical Integration of Function Handles

Now that we have function handles we can use integral to do numerical integration. However there's another caveat. The Matlab integral command does its work with vectors and consequently for reasons we will not go into right now we must replace * and / and ^ with .* and ./ and .^. For example:
clear all;
f = @(x) exp(x.^2);
integral(f,1,2)

ans =
14.9900

The matlabFunction Command

If you're confused or frustrated there is a command, matlabFunction (notice the weird capitalization) which converts symbolic functions into function handles. The following might be prettier to you:
clear all;
syms f(x);
f(x) = exp(x^2);
integral(matlabFunction(f(x)),1,2)

ans =
14.9900

Lastly the function handle can just be plugged directly in, for example:
clear all;
integral(@(x) sin(x.^2),1,2)

ans =
Plotting Function Handles

The `ezplot` command however works fine other than a warning:

```matlab
clear all;
f = @(x) cos(x)^2;
ezplot(f,[-pi,pi])
```

*Warning: Function failed to evaluate on array inputs; vectorizing the function may speed up its evaluation and avoid the need to loop over array elements.*

The warning is just a comment about the internal functioning of Matlab. You can eliminate it by replacing `^` with `.^` or by using a symbolic function instead.

More Complicated Calculations - IMPORTANT!

It's important to realize that if $f$ is a function handle and $x$ is symbolic then $f(x)$ and consequently things like `diff(f(x))` are symbolic too. This means we need to use `subs` in places. For example if we wish to find a second derivative of a function handle and then plug in:

```matlab
clear all;
```
f = @(x) 1/x^3+log(x-2);
syms x;
subs(diff(f(x),2),3)

\[ ans = \]
\[ -0.9506 \]

Take a moment to understand what this does, keeping in mind the fact that \( x \) is nested inside \( f(x) \) which is inside \( \text{diff} \) which is inside \( \text{subs} \).

Similarly suppose we wanted to start with a function handle, differentiate, divide by \( x \) and then plug in:

clear all;
f = @(x) x^3+2/x;
syms x;
subs(diff(f(x))/x,5)

\[ ans = \]
\[ 14.9840 \]

*Published with MATLAB® 8.0*