

### [7.16] Use *when()* instead of *if...then...else...endif*

The 92+ manual describes using *when()* only to create discontinuous graphs, however, it is much more useful than that. It can be used in place of the *if...endif* construction, and is more compact.

*when()* functions can also be nested to create an *if...then...else..endif* structure. Suppose you have four functions  $f_1(x)$ ,  $f_2(x)$ ,  $f_3(x)$  and  $f_4(x)$ . You want to evaluate the functions on intervals like this:

```
f1(x) when x < 1
f2(x) when x >= 1 and x < 2
f3(x) when x >= 2 and x < 3
f4(x) when x >= 3
```

The *if...Endif* version looks like this:

```
if x < 1 then
  f1(x)
elseif x >= 1 and x < 2 then
  f2(x)
elseif x >= 2 and x < 3 then
  f3(x)
else
  f4(x)
endif
```

The nested-*when()* version looks like this:

```
when(x < 1, f1(x), when(x < 2, f2(x), when(x < 3, f3(x), f4(x))))
```

The *if...endif* version is 107 bytes and executes in about 112 mS/call. The nested-when version is 73 bytes, and executes in about 100 mS/call. So, this method runs slightly faster and uses much less memory.