

[6.5] Round numbers to significant digits

You may need to round a number to a specified number of significant digits. For example, this can be useful when simulating arithmetic on a different processor with fewer significant digits, or estimating the effects of round-off error on function evaluation. The built-in function *round()* will not work, because it rounds to a specified number of digits after the decimal point. This function does it, though:

```
sigdig(x,n)
func
©(x,n) round x to n sig digits
©x is number, list, matrix
©n is 1 to 12
©13mar00/dburkett@infinet.com
local s,xlm,k,j,n1,n2

if gettype(x)="NUM" then

format(approx(x),"s12")→s
return
expr(format(round(expr(left(s,instring(s,"E")-1)),n),"f"&string(exact(n-1)))&right(s,dim(s)-instring(s,"E")+1))

elseif gettype(x)="LIST" then
dim(x)→n1
newlist(n1)→xlm
for k,1,n1
sigdig(x[k],n)→xlm[k]
endfor
return xlm

elseif gettype(x)="MAT" then
rowdim(x)→n1
coldim(x)→n2
newmat(n1,n2)→xlm
for j,1,n1
for k,1,n2
sigdig(x[j,k],n)→xlm[j,k]
endfor
endfor
return xlm

endif

Endfunc
```

Like the built-in *round()* function, *sigdig()* works on single numbers, lists or matrices. This is done by testing the type of the input argument *x*. If *x* is a number, it is rounded and returned. If *x* is a list or matrix, the individual elements are processed by a recursive calls to *sigdig()*.

The actual rounding is performed by using *round()* on the mantissa of the argument. This process is simplified by using *format()* to convert the input argument to scientific notation, which makes it easy to operate on the argument mantissa.

It would be an interesting challenge to modify this routine to work on complex numbers, in rectangular or polar format.