

between the three fit functions. Since the coefficient waves for the second and third terms set their respective y_0 to zero, this puts all the vertical offset into the one coefficient for the first term.

In this example the hold strings are literal quoted strings. They can be any string expression but see below for restrictions if you use a function list contained in a string:

```
String holdStr = "1"
FuncFit {{exp,expTerm1},{exp,expTerm2,hold=holdStr},
        {exp,expTerm3,hold="1"}} expSumData/D      // All on one line
```

The history report for a sum of fit functions is enhanced to show all the functions:

```
Fit converged properly
fit_expSumData= Sum of Functions(,x)
expTerm1={1.027,1.045,2.2288}
expTerm2={0,1.4446,10.416}
expTerm3={0,2.0156,102.28}
V_chisq= 0.864844; V_npnts= 1000; V_numNaNs= 0; V_numINFs= 0;
V_startRow= 0; V_endRow= 999; V_startCol= 0; V_endCol= 0;
V_startLayer= 0; V_endLayer= 0; V_startChunk= 0; V_endChunk= 0;
W_sigma={0.0184,0.0484,0.185,0,0.052,0.495,0,0.0239,2.34}
For function 1: exp:
Coefficient values  $\pm$  one standard deviation
  y0      = 1.027  $\pm$  0.0184
  A       = 1.045  $\pm$  0.0484
  invTau  = 2.2288  $\pm$  0.185
For function 2: exp:
Coefficient values  $\pm$  one standard deviation
  y0      = 0  $\pm$  0
  A       = 1.4446  $\pm$  0.052
  invTau  = 10.416  $\pm$  0.495
For function 3: exp:
Coefficient values  $\pm$  one standard deviation
  y0      = 0  $\pm$  0
  A       = 2.0156  $\pm$  0.0239
  invTau  = 102.28  $\pm$  2.34
```

Example: Function List in a String

The list of functions in the FuncFit command in the first example is moderately long, but it could be much longer. If you wanted to sum 20 peak functions and a baseline function, the list could easily exceed the limit of 2500 bytes in a command line.

Fortunately, you can build the function specification in a string variable and use the string keyword. Doing this on the command line for the first example looks like this:

```
String myFunctions="{exp, expTerm1}"
myFunctions+="{exp, expTerm2, hold=\"1\"}"
myFunctions+="{exp, expTerm3, hold=\"1\"}"
FuncFit {string = myFunctions} expSumData/D
```

These commands build the list of functions one function specification at a time. The second and third lines use the += assignment operator to add additional functions to the list. Each function specification includes its pair of braces.

Notice the treatment of the hold strings — in order to include quotation marks in a quoted string expression, you must escape the quotation marks. Otherwise the command line parser thinks that the first quote around the hold string is the closing quote.

The function list string is parsed at run-time outside the context in which FuncFit is running. Consequently, you cannot reference local variables in a user-defined function. The hold string may be either a quoted literal string, as shown here, or it can be a reference to a global variable, including the full data folder path: