```
    return cmplx(sn*xx/fn,sn*xx*pi/(fn*(n2+1)))
End

// Execute:
SumSeries series=s2,lowerLimit=0,upperLimit=INF
Print V_resultR,V_resultI
```

In the next example we sum 18 terms in the series for exp(x) using a parameter wave to pass the value of x.

```
Function fex(paramWave, index)
    Wave/Z paramWave       // Contains x value at which to evaluate exp(x)
    Variable index

    Variable xx = paramWave[0]
    if (index == 0)
        return 1
    endif
    return (xx^index) / factorial(index)
End

// Execute:
Make/D/O exponent = {-1}          // The value of x at which to evaluate exp(x)
SumSeries series=fex, lowerLimit=0, upperLimit=17, paramWave=exponent
Print/D V_resultR - exp(-1)       // Compare with built-in exp function
```

See Also

Integrate1D, **sum**

SVAR

SVAR [**/Z**][**/SDFR**=*dfr*] ***localName*** [**= *pathToStr***][**, *localName1* [**= *pathToStr1***]]…

SVAR is a declaration that creates a local reference to a global string variable accessed in a user-defined function.

The SVAR reference is required when you access a global string variable in a function. At compile time, the SVAR statement specifies a local name referencing a global string variable. At runtime, it makes the connection between the local name and the actual global variable. For this connection to be made, the global string variable must exist when the SVAR statement is executed.

When *localName* is the same as the global string variable name and you want to reference a global variable in the current data folder, you can omit *pathToStr*.

pathToStr can be a full literal path (e.g., root:FolderA:var0), a partial literal path (e.g., :FolderA:var0) or $ followed by string variable containing a computed path (see **Converting a String into a Reference Using $** on page IV-62).

You can also use a data folder reference or the /SDFR flag to specify the location of the string variable if it is not in the current data folder. See **Data Folder References** on page IV-78 and **The /SDFR Flag** on page IV-80 for details.

If the global variable may not exist at runtime, use the /Z flag and call **SVAR_Exists** before accessing the variable. The /Z flag prevents Igor from flagging a missing global variable as an error and dropping into the Igor debugger. For example:

```
SVAR/Z nv=<pathToPossiblyMissingStringVariable>
if( SVAR_Exists(sv) )
    <do something with sv>
endif
```

Note that to create a global string variable, you use the **String**/G operation.

Flags

/SDFR=*dfr*	Specifies the source data folder. See **The /SDFR Flag** on page IV-80 for details.
/Z	An SVAR reference to a null string variable does not cause an error or a debugger break.

See Also

SVAR_Exists function.

Accessing Global Variables and Waves on page IV-65.