```
SumDimension/D=1/DEST=wout wave4d
```

creates a wave wout that satisfies

$$wout[i][k][l] = \sum_{j=0}^{\dim 1-1} wave4d[i][j][k][l],$$

and wout has dimensions dim0 x dim2 x dim3.

If any values in *srcWave* are NaN, the corresponding sum element will be NaN.

See Also

sum

MatrixOp keywords sumRows, sumCols, sumBeams

ImageTransform keywords sumAllCols, sumAllRows, sumPlane, sumPlanes

SumSeries

SumSeries [*flags*] *keyword=value*

The SumSeries operation computes the sum of the results returned from a user-defined function for input values between two specified index limits.

SumSeries was added in Igor Pro 7.00.

Flags

/CCNT=*nc* When summing with one or two infinite limits you can use this flag to specify the minimum number of calls to the summand function which, when added to the sum, produce a change that is less than the tolerance. By default *nc*=10.

If you are summing a well-behaved monotonic series it is sufficient to set *nc*=1. In some pathological cases it is useful to check that the sum remains effectively unchanged even after many terms are added to the series.

/INAN Ignore NaNs returned from the user function. In the case of a complex valued summand, a NaN in either the real or imaginary components excludes the contribution of the term to the sum.

/Q Quiet mode; do not print in the history.

/Z[=z] /Z or /Z=1 prevents reporting any errors. If the operation encounters an error it sets V_Flag to the error code.

Keywords

lowerLimit=*n1* Specifies the starting index at which the summand is evaluated. *n1* must be either an integer -INF.

series=*userFunc* Specifies the name of the user function that returns the summand (i.e., a single term in the sum that corresponds to the input index). See **The SumSeries Summand Function** below for details.

upperLimit=*n2* Specifies the last value at which the summand is evaluated. *n2* must be either an integer INF.

tolerance=*tol* Specifies a tolerance value used when one or both of the limits are infinite. By default, the tolerance value is 1e-10. *tol* must be finite. If both limits are finite this keyword is ignored.

paramWave=*pw* *pw* is a single-precision or double-precision wave that is passed to the summand function. This is useful if you need to provide the summand function external/global data.

If you omit the paramWave keyword then the summand function receives a null wave as the parameter wave.

The SumSeries Summand Function

You specify the summand function using the series keyword. The form of the user-defined summand function is:

```
Function summandReal(inW,index)
    Wave inW
    Variable index
    ... compute something
    return result
End
```

The index changes by 1 for each successive call to the summand.

You can also define a complex summand function:

```
Function/C summandComplex(inW,index)
    Wave inW
    Variable index
    ... compute something
    Variable/C result
    return result
End
```

Details

The SumSeries operation is primarily intended for use with one or two infinite limits. If both limits are finite the operation performs the straightforward sum by calling the summand function once for every index from lowerLimit to upperLimit, inclusive.

If one limit is infinite the sum is evaluated by starting from the finite limit and proceeding in the direction of the infinite limit index until convergence is reached. Convergence in this context is defined as multiple (*nc*) consecutive calls to the summand which do not change the value of the sum by more than the tolerance value. By default *nc*=10 but you can change it using the /CCNT flag.

When both limits are infinite the operation first computes the sum for indices 0 to INF and then the sum from -1 to -INF. The two calculations are independent and require that the same convergence condition is met independently in each case. When the summand function is complex the convergence condition must hold for the real and imaginary components independently.

The operation does not perform any test on the summand function to estimate its rate of convergence. If you provide a non-converging summand function the operation can run indefinitely. You can abort it by pressing the **User Abort Key Combinations** or by clicking the Abort button.

The result of the sum is stored in V_resultR and, if the summand function returns a complex result, V_resultI.

If the calculation completes without error V_Flag is set to 0. Otherwise it contains an error code.

Examples

A simple test case is the geometric series for powers of 1/2. The sum of x^i for i=0 to i=INF where 0<x<1 is given by 1/(1-x). For x=1/2, this sum is 2.

```
Function s1(paramWave, index)
    Wave/Z paramWave        // Not used
    Variable index

    return 0.5^index
End

// Execute:
SumSeries series=s1,lowerLimit=0,upperLimit=INF
Print V_resultR
```

In the next example we use the series expansion of cosine and sine to evaluate exp(i*pi).

```
Function/C s2(paramWave, index)
    Wave/Z paramWave        // Not used
    Variable index

    Variable n2=2*index
    Variable xx=pi^n2
    Variable sn=(-1)^index
    Variable fn=Factorial(n2)
```