

MatrixOp Multithreading

Common CPUs are capable of running multiple threads. Some calculations are well suited to run in parallel. There are several ways to take advantage of multithreading using MatrixOp:

- User-created preemptive threads
MatrixOp is thread-safe so you can call it from preemptive threads. See **ThreadSafe Functions and Multitasking** on page IV-329 for details.
- Layer threads
If you are evaluating expressions that involve multiple layers you can use the /NTHR flag to run each layer calculation in a separate thread. When you account for thread overhead it makes sense to use /NTHR when the per-layer calculations are on the order of 1 million CPU cycles or more.
- Internal multithreading of operations or functions
Some MatrixOp functions are automatically multithreaded for SP and DP data. These include matrix-matrix multiplication, trigonometric functions, hypot, sqrt, erf, erfc, inverseErf, and inverseErfc. The **MultiThreadingControl** operation provides fine-tuning of automatic multithreading but you normally do not need to tinker with it.

MatrixOp Performance

In most situations MatrixOp is faster than a wave assignment statement or **FastOp**. However, for small waves the extra overhead may make it slower.

MatrixOp works fastest on floating point data types. For maximum speed, convert integer waves to single-precision floating point before calling MatrixOp.

Some MatrixOp expressions are evaluated with automatic multithreading. See **MatrixOp Multithreading** on page III-148 for details.

MatrixOp Optimization Examples

The section shows examples of using MatrixOp to improve performance.

- Replace matrix manipulation code with MatrixOp calls. For example, replace this:

```
Make/O/N=(vecSize,vecSize) identityMatrix = p==q ? 1 : 0
MatrixMultiply matB, matC
identityMatrix -= M_Product
MatrixMultiply identityMatrix, matD
MatrixInverse M_Product
Rename M_Inverse, matA
```

with:

```
MatrixOp matA = Inv((Identity(vecSize) - matB x matC) x matD)
```

- Replace waveform assignment statements with MatrixOp calls. For example, replace this:

```
Duplicate/O wave2,wave1
wave1 = wave2*2
```

with:

```
MatrixOp/O wave1 = wave2*2
```

- Factor and compute only once any repeated sub-expressions. For example, replace this:

```
MatrixOp/O wave1 = var1*wave2*wave3
MatrixOp/O wave4 = var2*wave2*wave3
```

with:

```
MatrixOp/O/FREE tmp = wave2*wave3 // Compute the product only once
MatrixOp/O wave1 = var1*tmp
MatrixOp/O wave4 = var2*tmp
```