Primitive Function Performance of APL2 Version 1 Release 3 (with SPE PL34409) on the IBM 3090/S Vector Facility

Document Number TR 21.1292

March 1989

Mario V. Morreale
Michael Van Der Meulen

IBM Corporation
Neighborhood Road
Kingston, New York 12401

First Edition (March 1989)

This edition applies to APL2 Version 1 Release 3 (with SPE PL34409) (Program Number 5668-899).

The information contained in this document has not been submitted to any formal IBM test and is distributed on an *as is* basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

The performance data contained in this document was obtained in a controlled environment based on the use of specific data and is presented only to illustrate techniques and procedures to assist personnel to better understand IBM products. The results that may be obtained in other operating environments may vary significantly. Users of this report should verify the applicable data in their specific environment.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

© Copyright International Business Machines Corporation, 1989

Abstract

This document reports the results of timings of the Vector Facility (VF) support available on an IBM 3090/S processor using APL2 Version 1 Release 3 (with SPE PL34409).

Included are execution times for selected primitive functions and expressions. The measurements were first taken with the VF utilization turned off and then again with the VF utilization turned on. The ratios **scalar÷vector** are also reported.

Locator Terms

- APL (A Programming Language)
- IBM 3090 Vector Facility (3090 VF)
- APL2 Version 1 Release 3
- APL2 Performance
- Vector and Scalar APL2 Function Performance
- Vector Processing
- Array Processing
- Scientific Programming
- Numerical Intensive Computing (NIC)

Preface

This Technical Report was compiled at the IBM Kingston, New York Numerically Intensive Computing (NIC) Center.

The primitives selected were based on prior timings done at the IBM Santa Teresa Laboratory under the guidance of the APL2 Development team. In addition, other primitives were selected based on the authors' intuition, general interest and curiosity.

The authors do not imply that all possible vector cases are included. For example only a few cases of inner product and outer product were timed. Other scalar functions should show improvement when applied to inner product and outer product and when applied to reduction.

A few cases involving scalar extension were selected and timed as were scalar string expressions. Other scalar functions may show improvement in these cases.

The authors wish to thank the Santa Teresa APL Development group for all the assistance given when questions arose on specific results, specifically to Nancy Wheeler and Dr. James Brown. Special thanks to Nancy for her careful review of the content with respect to its completeness and accuracy. In addition to the above, we wish to thank Jon McGrew for his assistance in formatting the document.

Additional References

1. Brown, Dr. James
   "An APL Description of The IBM 3090 Vector Facility"
   Proceedings of ACM's APL88 Conference

2. Van Der Meulen, Michael, and Mario V. Morreale
   "APL2 Version 1 Release 3 Primitive Function Performance on the IBM 3090 Vector Facility"
   TR 21.1078, IBM Corporation, Kingston, NY
   April 1988

The APL2 Release 3 Library

Product Brochure— APL2: The Added Dimension	GC26-4218
An Introduction to APL2	SH20-9229
APL2 General Information	GH20-9214
APL2 GRAPHPAK: User's Guide and Reference	SH20-9230
APL2 Installation and Customization under CMS	SH20-9221
APL2 Installation and Customization under TSO	SH20-9222
APL2 Licensed Program Specifications	GH20-9213
APL2 Messages and Codes	SH20-9220
APL2 Migration Guide	SH20-9215
APL2 Programming: Guide	SH20-9216
APL2 Programming: Language Reference	SH20-9227
APL2 Programming: System Services Reference	SH20-9218
APL2 Programming: Processor Interface Reference	SH20-9234
APL2 Programming: Using Structured Query Language (SQL)	SH20-9217
APL2 Programming: Using the Supplied Routines	SH20-9233
APL2 Reference Card	SX26-3738
APL2 Reference Summary	SX26-3737
APL2 Diagnosis	LY27-9532
APL2 Directory of Programming Interfaces for Customers	GC26-4521

Table of Contents

Introduction

In February of 1988 APL2 Version 1 Release 3 was shipped to customers. Among other notable enhancements, it contained the first phase of support for the 3090 Vector Facility (VF). About a month later TR 21.1078 was published by the Kingston NIC Center to report performance data for the APL2 primitive functions and operators that were vectorized in that release.

In April of 1988 two APL2 performance PTF's were released for Vector Facility users. Documented in an addendum to TR 21.1078, PL25618 dramatically improved the VF performance of matrix multiplication, while PL27058 improved the VF performance of certain cases of the trigonometric tangent function.

Near the end of 1988 IBM introduced a new line of high-performance mainframes, the 3090/S processor family. Performance improvements included a reduction in processor cycle time to 15 nanoseconds, a doubling of cache from 64K to 128K, larger memories, and a new Vector Facility. The 3090/S VF doubled the section size (the length of the "pipe" used in pipelining arithmetic operations on strings of data) from 128 to 256 data elements. Also the new Vector Facility substantially improved the performance of divide operations.

The second phase of APL2 exploitation of the 3090 Vector Facility came in February of 1989, with the release of a very large SPE (Small Program Enhancement... a slight contradiction, like "jumbo shrimp"). SPE PL34409 includes VF support for many new primitive functions, and extends the domain of most other vectorized routines to include additional data types. In particular, the SPE adds boolean data support, including vectorization of all logical and relational functions.

The availability of the new 3090/S processor, enhanced Vector Facility, and additional VF support provided by APL2 SPE PL34409 prompted the authors to produce this new technical report on APL2 VF performance. While the faster processor and increased cache size of course improved scalar execution times as well, the authors were pleased to find that scalar/vector ratios improved an average of 12-15% for most vectorized primitives. Algorithm changes in the SPE accounted for even larger improvements in vector operations involving scalar extension, while the new VF hardware took credit for impressive scalar/vector ratios for division and reciprocal. In a few cases of complex arithmetic, particularly those involving stride, there were no improvements recorded in APL2 vector performance.

While Vector Facility performance improvements of 2 or 3 times that of scalar execution at the primitive function level are not uncommon, it should be noted that the timings in this paper should not be construed to imply comparable improvements when entire applications are timed. Performance of an application is dependent on many factors, and the vectorized primitives are only part of the application. The *TIME* facility in APL2 Release 3 often provides quick insight in determining the percentage of application CPU time spent doing numerically intensive computing, the portion most apt to benefit from the Vector Facility.

APL2 Version 1 Release 3 and SPE PL34409 include many other performance enhancements, in addition to Vector Facility exploitation. For more information, see Appendix B.

Timing Methodology

The basic timing method used was the creation of a function (***CODE***) for each statement being timed. Prior to execution, garbage collection is forced and the current setting of the CPU clock is held. (Garbage collection is a term used when the interpreter makes available memory that was used temporarily to store objects that are no longer needed for subsequent execution.)

The number of iterations chosen for each statement being timed varied according to the amount of CPU time consumed. A large number of iterations was chosen to compensate for the resolution of the ***⎕AI*** CPU counter when the time to execute the statement in question was extremely small. A small number of iterations was chosen when the execution time for the statement in question was considerable (see inner product of large matrices).

In all cases, the number of iterations chosen did not cause garbage collection during the timing of the statement. This was possible by first issuing the ***)CHECK TRACE SERVICE*** command just prior to trial runs and reviewing the session log for cases of garbage collection occurring within the ***CODE*** function. When garbage collection occurred the number of iterations was reduced. This process was repeated until garbage collection no longer occurred during the execution of the ***CODE*** function.

The statements were first executed with the vector hardware utilization turned off and then executed with the vector hardware utilization turned on.

An example showing the ***CODE*** function created and executed for the conjugate primitive "+" is shown on the next page:

```
      ∇
[0]     Z←CODE;T
[1]     T←GCOL
[2]     T←⎕AI[2]
[3]     Z←+C100
[4]     Z←+C100
[5]     Z←+C100
[6]     Z←+C100
[7]     Z←+C100
[8]     Z←+C100
[9]     Z←+C100
[10]    Z←+C100
[11]    Z←+C100
[12]    Z←+C100
[13]    Z←⎕AI[2]-T
      ∇ 01/04/1988 14.10.11 (GMT-5)
```

Notes:

- *GCOL* in statement 1 is a locked function. It initiates garbage collection.
- The statement is iterated and executed 10 times.
- In statements 3 through 12, *C*100 implies a complex vector of length 100.

Environment

The timings were performed in March 1989 on the Numerically Intensive Computing (NIC) MVS system located in Kingston, New York. APL2 was invoked using a workspace size of 30 megabytes.

Hardware

- IBM 3090-200/S with two Vector Facilities
- Real storage: 256 megabytes
- Expanded Storage: 512 megabytes

Software

- Operating System: MVS/ESA SP3.1.0
- TSO/E Version 1 Release 4
- APL2 Version 1 Release 3 (with SPE PL34409)

  The PTF numbers for APL2 (APAR PL34409) under MVS/TSO are:

  - UL90239
  - UL90241
  - UL90247
  - UL90248
  - UL90249
  - UL90250

  Note: the PTF numbers for APL2 (APAR PL34409) under VM/CMS are UL90238 and UL90240.

Summary Results

The naming convention used in the results that follow is a one-character prefix identifying the data type followed by an indication of rank. The prefix "*I*" implies integer, the prefix "*R*" real, the prefix "*C*" complex and the prefix "*B*" implies boolean. "*VECTOR*" implies a rank of 1, "*MATRIX*" implies a rank of 2, and "*ARRAY*" is used for three-dimensional (or higher rank) arrays.

The lengths used during the timing operations were:

- *VECTOR*:
  - 100, 200, 400, 800, 1600, 3200, 12800
- *MATRIX*:
  - 10×10, 20×20, 50×50, 100×2,
  - 100×100, 200×200, 400×400, 500×500
- *ARRAY*:
  - 10×2×2, 100×2×2, 500×2×2
  - 10×5×5, 100×5×5, 500×5×5
  - 10×10×10, 100×10×10, 500×10×10
  - 10×20×20, 100×20×20, 500×20×20
  - 2×10×2, 2×100×2, 2×500×2
  - 5×10×5, 5×100×5, 5×500×5
  - 10×10×10, 10×100×10, 10×500×10
  - 20×10×20, 20×100×20, 20×500×20
  - 2×2×10, 2×2×100, 2×2×500
  - 5×5×10, 5×5×100, 5×5×500
  - 10×10×10, 10×10×100, 10×10×500
  - 20×20×10, 20×20×100, 20×20×500

For more detail on the actual expression timed refer to the next section ("Detail Results" on page 21).

Monadic Scalar Primitives

	Scalar÷Vector Ratios	
Statement	Range	Average
+CVECTOR	1.5 - 1.9	1.7
-BVECTOR	1.6 - 4.4	3.3
-IVECTOR	1.7 - 3.8	3.0
-RVECTOR	1.7 - 3.1	2.6
-CVECTOR	1.5 - 2.1	1.9
×IVECTOR	1.3 - 2.1	1.9
×RVECTOR	1.7 - 3.0	2.5
×CVECTOR	2.6 - 3.0	2.9
÷BVECTOR	2.1 - 5.2	4.2
÷IVECTOR	3.6 - 6.0	5.3
÷RVECTOR	2.7 - 4.5	4.0
÷CVECTOR	3.3 - 3.8	3.7
⍟BVECTOR	2.0 - 5.0	3.9
⍟IVECTOR	2.9 - 3.8	3.6
⍟RVECTOR	2.5 - 3.5	3.2
⋆BVECTOR	2.0 - 3.9	3.2
⋆IVECTOR	2.9 - 3.9	3.6
⋆RVECTOR	2.5 - 3.4	3.3
⋆CVECTOR	2.4 - 2.9	2.8
○BVECTOR	1.7 - 3.6	3.0
○IVECTOR	2.8 - 5.6	4.8
○RVECTOR	1.7 - 3.2	2.7
○CVECTOR	1.6 - 2.2	2.0
|IVECTOR	1.7 - 3.7	3.0
|RVECTOR	1.7 - 3.0	2.6
|CVECTOR	2.3 - 2.8	2.7
?BVECTOR	1.7 - 5.1	3.7

Dyadic Scalar Primitives

	Scalar÷Vector Ratios	
Statement	*Range*	*Average*
IVECTOR+IVECTOR	1.0 - 1.7	1.4
SCALAR+IVECTOR	1.1 - 2.0	1.6
RVECTOR+RVECTOR	1.2 - 1.8	1.5
SCALAR+RVECTOR	1.2 - 2.0	1.7
CVECTOR+CVECTOR	1.1 - 1.4	1.3
SCALAR+CVECTOR	1.2 - 1.6	1.5
IVECTOR-IVECTOR	1.1 - 1.7	1.4
SCALAR-IVECTOR	1.1 - 2.0	1.6
RVECTOR-RVECTOR	1.2 - 1.8	1.5
SCALAR-RVECTOR	1.2 - 2.1	1.7
CVECTOR-CVECTOR	1.1 - 1.4	1.3
SCALAR-CVECTOR	1.2 - 1.7	1.5
BVECTOR×BVECTOR	.9 - 1.4	1.1
IVECTOR×IVECTOR	1.0 - 1.4	1.3
SCALAR×IVECTOR	1.1 - 1.5	1.4
RVECTOR×RVECTOR	1.2 - 1.9	1.6
SCALAR×RVECTOR	1.2 - 2.3	1.9
CVECTOR×CVECTOR	1.3 - 1.7	1.6
SCALAR×CVECTOR	1.3 - 1.9	1.7
SCALAR÷IVECTOR	2.3 - 4.7	4.0
RVECTOR÷RVECTOR	1.8 - 2.8	2.6
SCALAR÷RVECTOR	1.9 - 2.9	2.7
CVECTOR÷CVECTOR	3.1 - 3.8	3.7
SCALAR÷CVECTOR	3.3 - 4.0	3.8

Dyadic Scalar Primitives (continued)

	Scalar÷Vector Ratios	
Statement	Range	Average
BVECTOR⌊BVECTOR	.9 - 1.3	1.1
IVECTOR⌊IVECTOR	1.2 - 2.1	1.8
RVECTOR⌊RVECTOR	1.3 - 2.1	1.8
BVECTOR⌈BVECTOR	.9 - 1.4	1.1
IVECTOR⌈IVECTOR	1.2 - 2.1	1.8
RVECTOR⌈RVECTOR	1.2 - 2.1	1.8
BVECTOR⋆BVECTOR	1.1 - 2.2	1.6
RVECTOR⋆RVECTOR	2.5 - 3.0	2.8
IVECTOR⋆.5	2.1 - 3.1	2.8
RVECTOR⋆.5	3.0 - 3.7	3.5
IVECTOR⋆2	1.3 - 2.0	1.8
RVECTOR⋆2	1.1 - 1.7	1.5
10⍟IVECTOR	2.7 - 3.5	3.3
10⍟RVECTOR	2.5 - 3.2	3.1
BVECTOR=BVECTOR	.9 - 1.6	1.2
IVECTOR=IVECTOR	1.2 - 3.1	2.3
RVECTOR=RVECTOR	2.2 - 3.8	3.3
BVECTOR≠BVECTOR	.9 - 1.4	1.2
IVECTOR≠IVECTOR	1.2 - 3.1	2.3
RVECTOR≠RVECTOR	2.0 - 3.4	3.0
BVECTOR<BVECTOR	1.1 - 2.4	1.6
IVECTOR<IVECTOR	1.3 - 3.6	2.6
RVECTOR<RVECTOR	1.3 - 2.3	2.0
BVECTOR>BVECTOR	1.0 - 2.5	1.7
IVECTOR>IVECTOR	1.3 - 3.6	2.6
RVECTOR>RVECTOR	1.5 - 2.3	2.0

Dyadic Scalar Primitives (continued)

Statement	Scalar÷Vector Ratios Range	 Average
BVECTOR≤BVECTOR	1.0 - 2.5	1.7
IVECTOR≤IVECTOR	1.3 - 3.6	2.6
RVECTOR≤RVECTOR	1.6 - 2.4	2.1
BVECTOR≥BVECTOR	1.1 - 2.5	1.7
IVECTOR≥IVECTOR	1.3 - 3.6	2.6
RVECTOR≥RVECTOR	1.4 - 2.4	2.1
BVECTOR|BVECTOR	1.0 - 2.0	1.5
BVECTOR!BVECTOR	1.1 - 2.6	1.7
BVECTOR∧BVECTOR	1.0 - 1.3	1.2
BVECTOR∨BVECTOR	.9 - 1.4	1.1
BVECTOR⍲BVECTOR	1.0 - 1.5	1.2
BVECTOR⍱BVECTOR	.9 - 1.5	1.2

Circular Functions

	Scalar÷Vector Ratios	
Statement	Range	Average
0○*BVECTOR*	1.5 - 3.4	2.8
0○*RVECTOR*	3.0 - 3.9	3.7
1○*BVECTOR*	1.5 - 3.7	2.8
1○*IVECTOR*	2.4 - 3.1	2.9
1○*RVECTOR*	1.5 - 2.6	2.2
2○*BVECTOR*	1.6 - 3.4	2.7
2○*IVECTOR*	2.2 - 2.9	2.7
2○*RVECTOR*	1.5 - 2.5	2.2
3○*BVECTOR*	1.6 - 3.5	2.8
3○*IVECTOR*	1.4 - 1.6	1.6
3○*RVECTOR*	1.3 - 2.0	1.8
4○*BVECTOR*	1.5 - 3.4	2.7
4○*IVECTOR*	3.1 - 4.1	3.9
4○*RVECTOR*	2.9 - 3.8	3.6
5○*BVECTOR*	1.5 - 3.5	2.7
6○*BVECTOR*	1.5 - 3.3	2.7
7○*BVECTOR*	1.5 - 3.5	2.8
8○*BVECTOR*	1.3 - 2.1	1.8
9○*CVECTOR*	1.7 - 3.4	2.7
10○*IVECTOR*	1.7 - 3.7	2.9
10○*RVECTOR*	1.5 - 3.0	2.5
10○*CVECTOR*	2.3 - 2.8	2.7
11○*IVECTOR*	1.7 - 4.2	3.3
11○*RVECTOR*	1.7 - 3.4	2.7
11○*CVECTOR*	1.4 - 1.8	1.7
12○*BVECTOR*	1.6 - 3.5	2.8
12○*IVECTOR*	2.5 - 4.9	4.1
12○*RVECTOR*	1.7 - 3.0	2.5

Circular Functions (continued)

	Scalar÷Vector Ratios	
Statement	*Range*	*Average*
¯1○*BVECTOR*	1.5 - 3.4	2.8
¯2○*BVECTOR*	1.5 - 3.4	2.7
¯3○*BVECTOR*	1.5 - 3.5	2.8
¯3○*IVECTOR*	1.2 - 1.3	1.3
¯3○*RVECTOR*	1.2 - 1.9	1.6
¯4○*BVECTOR*	1.2 - 2.0	1.7
¯4○*IVECTOR*	3.2 - 4.2	3.9
¯4○*RVECTOR*	3.0 - 3.9	3.6
¯5○*BVECTOR*	1.5 - 3.3	2.7
¯6○*BVECTOR*	1.1 - 2.0	1.6
¯6○*IVECTOR*	3.2 - 4.0	3.7
¯6○*RVECTOR*	3.0 - 3.8	3.6
¯7○*BVECTOR*	1.9 - 4.8	3.7
¯8○*BVECTOR*	1.3 - 2.0	1.7
¯10○*CVECTOR*	1.3 - 1.7	1.6
¯11○*BVECTOR*	1.3 - 2.0	1.8
¯11○*IVECTOR*	2.0 - 2.8	2.6
¯11○*RVECTOR*	1.7 - 2.2	2.0
¯11○*CVECTOR*	1.4 - 1.8	1.7
¯12○*BVECTOR*	1.3 - 2.0	1.8
¯12○*IVECTOR*	2.3 - 2.7	2.6
¯12○*RVECTOR*	1.8 - 2.5	2.3
¯12○*CVECTOR*	2.6 - 2.9	2.9

Non-Scalar Primitives

	Scalar÷Vector Ratios			
Statement	*Range*			*Average*
+/*BVECTOR*	1.1	-	5.2	2.8
+/*RVECTOR*	1.0	-	2.1	1.6
+/[1]*IMATRIX*	2.2	-	2.4	2.3
+/[2]*IMATRIX*	1.0	-	2.2	1.4
+/[1]*RMATRIX*	2.2	-	4.0	2.8
+/[2]*RMATRIX*	1.5	-	2.1	1.8
÷/[2]*IMATRIX*	3.0	-	8.4	6.7
÷/[2]*RMATRIX*	3.3	-	6.3	5.5
+/[1]*IARRAY*	1.0	-	4.7	2.0
+/[1]*RARRAY*	1.0	-	4.8	2.4
+/[1]*CARRAY*	.9	-	3.0	1.6
+/[2]*IARRAY*	1.0	-	2.5	1.1
+/[2]*RARRAY*	1.0	-	2.5	1.4
+/[2]*CARRAY*	.9	-	1.4	1.0
+/[3]*IARRAY*	1.0	-	4.7	1.8
+/[3]*RARRAY*	1.0	-	4.6	2.0
+/[3]*CARRAY*	1.0	-	3.3	1.3
IMATRIX+.×*IMATRIX*	1.3	-	1.5	1.4
RMATRIX+.×*RMATRIX*	8.0	-	10.7	9.0
CMATRIX+.×*CMATRIX*	2.6	-	3.2	2.9
IMATRIX+.×*BMATRIX*	8.7	-	37.3	26.1
BMATRIX+.×*IMATRIX*	12.6	-	58.3	40.3
RMATRIX+.×*BMATRIX*	8.1	-	34.5	24.1
BMATRIX+.×*RMATRIX*	15.0	-	56.1	38.4

Non-Scalar Primitives (continued)

	Scalar÷Vector Ratios	
Statement	*Range*	*Average*
IVECTOR∘.×*IVECTOR*	1.5 - 1.7	1.6
RVECTOR∘.×*RVECTOR*	2.0 - 2.7	2.3
CVECTOR∘.×*CVECTOR*	2.0 - 2.1	2.0
IVECTOR∘.+*IVECTOR*	1.5 - 2.2	1.9
RVECTOR∘.+*RVECTOR*	1.9 - 2.4	2.1
CVECTOR∘.+*CVECTOR*	1.5 - 1.8	1.7
BVECTOR\IVECTOR	3.7 - 6.6	5.6
BVECTOR\RVECTOR	4.0 - 5.9	5.3
BVECTOR\CVECTOR	3.7 - 4.9	4.2
BVECTOR,*IVECTOR*	1.3 - 3.0	2.3
BVECTOR,*RVECTOR*	1.2 - 2.2	1.8
BVECTOR,*CVECTOR*	1.4 - 2.0	1.7
IVECTOR,*RVECTOR*	1.8 - 2.5	2.1
IVECTOR,*CVECTOR*	1.5 - 1.8	1.6
RVECTOR,*CVECTOR*	1.0 - 1.5	1.2
IPROGRESS,*IVECTOR*	1.0 - 1.6	1.3
IPROGRESS,*RVECTOR*	1.0 - 1.2	1.1
IPROGRESS,*CVECTOR*	1.1 - 1.2	1.2
⍉*IMATRIX*	1.3 - 2.3	1.9
⍉*RMATRIX*	1.0 - 1.8	1.5
⍉*CMATRIX*	1.1 - 2.3	1.5
⌽[2]*IMATRIX*	1.2 - 2.2	1.8
⌽[2]*RMATRIX*	1.0 - 2.1	1.6
⌽[2]*CMATRIX*	1.0 - 1.5	1.3

Non-Scalar Primitives (continued)

Statement	Scalar÷Vector Ratios	
	Range	Average
2-/*IVECTOR*	4.3 - 14.2	10.9
2-/*RVECTOR*	4.3 - 11.6	8.9
2-/[1]*IMATRIX*	11.6 - 13.5	12.4
2-/[2]*IMATRIX*	14.2 - 15.2	14.7
2-/[1]*RMATRIX*	9.4 - 10.8	9.9
2-/[2]*RMATRIX*	11.8 - 12.6	12.1
2+/*IVECTOR*	4.6 - 15.4	11.1
2+/*RVECTOR*	4.2 - 11.6	8.9
2+/[1]*IMATRIX*	11.6 - 13.5	12.5
2+/[2]*IMATRIX*	14.6 - 16.2	15.1
2+/[2]*RMATRIX*	9.4 - 10.3	9.8
2+/[2]*RMATRIX*	10.7 - 13.7	11.7
*I*12800[*IVECTOR*]	1.0 - 2.1	1.6
*R*12800[*IVECTOR*]	.9 - 1.8	1.4
IMATRIX[;5]	3.1 - 5.9	4.3
RMATRIX[;5]	2.9 - 4.3	3.8
IMATRIX[;*IVECTOR*]	1.2 - 2.7	2.1
RMATRIX[;*IVECTOR*]	1.1 - 3.1	2.0
*I*12800[*IVECTOR*]←1234	1.0 - 3.3	2.0
*R*12800[*IVECTOR*]←.1234	1.0 - 3.3	2.0

Idioms

Statement	Scalar÷Vector Ratios Range	Average
+/[1]|*IMATRIX*	2.9 - 3.4	3.1
+/[2]|*IMATRIX*	2.1 - 3.4	2.5
+/[1]|*RMATRIX*	2.5 - 3.2	2.9
+/[2]|*RMATRIX*	2.4 - 2.7	2.5
+/[1]|*CMATRIX*	2.8 - 2.8	2.8
+/[2]|*CMATRIX*	2.7 - 2.8	2.8
⌈/[1]|*IMATRIX*	2.9 - 3.4	3.1
⌈/[2]|*IMATRIX*	1.9 - 3.5	2.4
⌈/[1]|*RMATRIX*	2.5 - 3.1	2.8
⌈/[2]|*RMATRIX*	2.1 - 2.3	2.2
⌈/[1]|*CMATRIX*	2.8 - 2.8	2.8
⌈/[2]|*CMATRIK*	2.7 - 2.7	2.7
⌊/[1]|*IMATRIX*	2.7 - 3.2	2.9
⌊/[2]|*IMATRIX*	2.0 - 3.3	2.4
⌊/[1]|*RMATRIX*	2.6 - 3.1	2.9
⌊/[2]|*RMATRIX*	2.0 - 2.2	2.1
⌊/[1]|*CMATRIX*	2.8 - 2.8	2.8
⌊/[2]|*CMATRIX*	2.7 - 2.7	2.7
RVECTOR⍳⌈/*RVECTOR*	1.0 - 1.1	1.1
RVECTOR⍳⌊/*RVECTOR*	1.1 - 1.3	1.2
⍎1⍕*VECTOR*	1.1 - 2.1	1.9
⍎2⍕*VECTOR*	1.1 - 2.1	1.9
⍎3⍕*VECTOR*	1.1 - 2.2	1.9
⍎4⍕*VECTOR*	1.1 - 2.1	1.9
⍎5⍕*VECTOR*	1.1 - 2.1	1.9

External Function ATR (Array to Record)

	Scalar÷Vector Ratios	
Statement	Range	Average
I2 ATR I4	1.1 - 2.3	1.6
E16 ATR I4	1.2 - 1.9	1.5
J8 ATR I4	1.2 - 2.3	1.8
J32 ATR I4	1.1 - 2.0	1.4
E16 ATR E8	1.0 - 1.4	1.2
J32 ATR E8	1.0 - 1.6	1.2
J32 ATR J16	1.0 - 1.7	1.2
E4 ATR BOOLEAN	1.1 - 3.0	1.9
E16 ATR BOOLEAN	1.1 - 1.7	1.4
J8 ATR BOOLEAN	1.1 - 2.7	1.9
J32 ATR BOOLEAN	1.0 - 1.7	1.3
E4 ATR PROGRESSION	1.0 - 1.3	1.1
E16 ATR PROGRESSION	1.0 - 1.3	1.2
J8 ATR PROGRESSION	1.0 - 1.4	1.2
J32 ATR PROGRESSION	1.0 - 1.8	1.2

Strings of Scalar Functions

With Loop Optimization

Statement	Scalar÷Vector Ratios Range	 Average
3.1-A1+B1+2.1×A1÷1.12-A1⌊B1×3.1	2.4 - 5.3	4.1

Without Loop Optimization

Statement	Scalar÷Vector Ratios Range	 Average
3.1-(A1+(B1+(2.1×(A1÷(1.12-(A1⌊(B1×3.1)))))))	1.3 - 2.1	1.8

For additional information, refer to "Strings of Scalar Functions" on page 88.

Detail Results

Note: All scalar and vector times reported in this section are average CPU times in milliseconds for *one* execution of the APL statement shown.

Monadic Scalar Primitives

Iterations	Statement	Scalar	Vector	Ratio
3000	+C100	.052	.034	1.52
3000	+C200	.084	.044	1.92
1000	+C400	.148	.087	1.70
200	+C800	.360	.230	1.57
100	+C1600	.670	.400	1.68
50	+C3200	1.380	.820	1.68
50	+C6400	2.800	1.640	1.71
50	+C12800	5.460	3.180	1.72
			Average	1.69

Iterations	Statement	Scalar	Vector	Ratio
2000	-B100M	.045	.028	1.64
2000	-B200M	.070	.032	2.22
1000	-B400M	.120	.043	2.79
500	-B800M	.228	.064	3.56
100	-B1600M	.440	.100	4.40
100	-B3200M	.930	.250	3.72
100	-B6400M	1.820	.440	4.14
100	-B12800M	3.580	.830	4.31
			Average	3.35

Iterations	Statement	Scalar	Vector	Ratio
3000	-I100	.045	.027	1.67
3000	-I200	.070	.032	2.22
2000	-I400	.127	.046	2.78
500	-I800	.232	.072	3.22
100	-I1600	.450	.120	3.75
100	-I3200	.950	.290	3.28
100	-I6400	1.890	.540	3.50
100	-I12800	3.760	1.070	3.51
			Average	2.99

Iterations	Statement	Scalar	Vector	Ratio
3000	-R100	.045	.027	1.69
3000	-R200	.071	.031	2.26
2000	-R400	.129	.049	2.63
500	-R800	.238	.078	3.05
100	-R1600	.540	.200	2.70
100	-R3200	1.040	.370	2.81
100	-R6400	2.060	.730	2.82
100	-R12800	4.170	1.420	2.94
			Average	2.61

Iterations	Statement	Scalar	Vector	Ratio
3000	−C100	.053	.035	1.52
3000	−C200	.095	.045	2.10
1000	−C400	.171	.089	1.92
200	−C800	.405	.230	1.76
100	−C1600	.770	.420	1.83
50	−C3200	1.560	.820	1.90
50	−C6400	3.200	1.700	1.88
50	−C12800	6.260	3.220	1.94
			Average	1.86

Iterations	Statement	Scalar	Vector	Ratio
3000	×I100	.048	.035	1.35
3000	×I200	.077	.047	1.64
2000	×I400	.138	.075	1.83
500	×I800	.256	.130	1.97
200	×I1600	.495	.235	2.11
200	×I3200	1.050	.530	1.98
100	×I6400	2.050	.980	2.09
100	×I12800	4.100	1.950	2.10
			Average	1.88

Iterations	Statement	Scalar	Vector	Ratio
3000	×R100	.059	.035	1.70
3000	×R200	.100	.046	2.17
2000	×R400	.183	.074	2.47
500	×R800	.348	.128	2.72
200	×R1600	.680	.230	2.96
200	×R3200	1.430	.530	2.70
100	×R6400	2.840	1.020	2.78
100	×R12800	5.790	2.020	2.87
			Average	2.55

Iterations	Statement	Scalar	Vector	Ratio
500	×C100	.540	.210	2.57
500	×C200	1.062	.360	2.95
200	×C400	2.110	.710	2.97
50	×C800	4.280	1.480	2.89
50	×C1600	8.520	2.880	2.96
25	×C3200	17.080	5.720	2.99
25	×C6400	34.120	11.280	3.02
25	×C12800	68.120	22.520	3.02
			Average	2.92

Iterations	Statement	Scalar	Vector	Ratio
1000	+B1000	.052	.025	2.08
500	+B2000	.084	.028	3.00
400	+B4000	.160	.040	4.00
300	+B8000	.297	.057	5.24
100	+B16000	.650	.160	4.06
50	+B32000	1.260	.260	4.85
50	+B64000	2.460	.480	5.13
25	+B128000	4.880	.960	5.08
			Average	4.18

Iterations	Statement	Scalar	Vector	Ratio
2000	+I100	.131	.036	3.63
2000	+I200	.241	.049	4.91
1000	+I400	.463	.085	5.45
500	+I800	.904	.150	6.03
400	+I1600	1.873	.348	5.39
300	+I3200	3.693	.643	5.74
200	+I6400	7.370	1.260	5.85
100	+I12800	14.760	2.540	5.81
			Average	5.35

Iterations	Statement	Scalar	Vector	Ratio
2000	+R100	.092	.034	2.75
2000	+R200	.166	.045	3.73
1000	+R400	.314	.076	4.13
500	+R800	.606	.134	4.52
400	+R1600	1.275	.315	4.05
300	+R3200	2.503	.590	4.24
200	+R6400	5.025	1.185	4.24
100	+R12800	10.090	2.320	4.35
			Average	4.00

Iterations	Statement	Scalar	Vector	Ratio
200	+C100	.360	.110	3.27
200	+C200	.700	.190	3.68
100	+C400	1.400	.370	3.78
50	+C800	2.880	.800	3.60
50	+C1600	5.680	1.520	3.74
25	+C3200	11.360	3.040	3.74
25	+C6400	22.760	6.080	3.74
20	+C12800	45.400	12.050	3.77
			Average	3.67

Iterations	Statement	Scalar	Vector	Ratio
75	●B1000	.053	.027	2.00
75	●B2000	.093	.040	2.33
50	●B4000	.180	.040	4.50
50	●B8000	.300	.060	5.00
25	●B16000	.640	.160	4.00
10	●B32000	1.200	.300	4.00
10	●B64000	2.400	.500	4.80
10	●B128000	4.900	1.000	4.90
			Average	3.94

Iterations	Statement	Scalar	Vector	Ratio
100	●I100	.290	.100	2.90
100	●I200	.550	.160	3.44
50	●I400	1.080	.300	3.60
50	●I800	2.140	.580	3.69
25	●I1600	4.360	1.200	3.63
10	●I3200	8.600	2.300	3.74
10	●I6400	17.300	4.500	3.84
10	●I12800	34.600	9.000	3.84
			Average	3.59

Iterations	Statement	Scalar	Vector	Ratio
100	●R100	.250	.100	2.50
100	●R200	.480	.160	3.00
50	●R400	.940	.300	3.13
50	●R800	1.880	.540	3.48
25	●R1600	3.840	1.160	3.31
10	●R3200	7.600	2.300	3.30
10	●R6400	15.200	4.400	3.45
10	●R12800	30.400	8.800	3.45
			Average	3.20

Iterations	Statement	Scalar	Vector	Ratio
75	*B100M	.053	.027	2.00
75	*B200M	.080	.027	3.00
50	*B400M	.140	.040	3.50
25	*B800M	.280	.080	3.50
25	*B1600M	.560	.200	2.80
20	*B3200M	1.050	.350	3.00
20	*B6400M	2.150	.550	3.91
10	*B12800M	4.200	1.200	3.50
			Average	3.15

Iterations	Statement	Scalar	Vector	Ratio
100	*I100	.290	.100	2.90
75	*I200	.547	.160	3.42
50	*I400	1.100	.300	3.67
25	*I800	2.160	.560	3.86
25	*I1600	4.400	1.160	3.79
20	*I3200	8.750	2.300	3.80
20	*I6400	17.500	4.550	3.85
10	*I12800	35.000	9.200	3.80
			Average	3.64

Iterations	Statement	Scalar	Vector	Ratio
100	*R100	.250	.100	2.50
75	*R200	.480	.147	3.27
50	*R400	.960	.300	3.20
25	*R800	1.880	.560	3.36
25	*R1600	3.840	1.120	3.43
20	*R3200	7.700	2.250	3.42
20	*R6400	15.300	4.500	3.40
10	*R12800	30.700	8.900	3.45
			Average	3.25

Iterations	Statement	Scalar	Vector	Ratio
50	*C100	.760	.320	2.38
50	*C200	1.500	.540	2.78
25	*C400	3.040	1.080	2.81
25	*C800	6.120	2.240	2.73
25	*C1600	12.200	4.320	2.82
25	*C3200	24.360	8.480	2.87
10	*C6400	48.600	16.700	2.91
10	*C12800	97.200	33.400	2.91
			Average	2.78

Iterations	Statement	Scalar	Vector	Ratio
1000	oB100M	.047	.028	1.68
1000	oB200M	.075	.033	2.27
1000	oB400M	.137	.048	2.85
500	oB800M	.256	.072	3.56
500	oB1600M	.568	.188	3.02
300	oB3200M	1.097	.317	3.46
200	oB6400M	2.130	.590	3.61
100	oB12800M	4.240	1.190	3.56
			Average	3.00

Iterations	Statement	Scalar	Vector	Ratio
3000	oI100	.082	.029	2.82
3000	oI200	.145	.036	4.07
1000	oI400	.278	.057	4.88
500	oI800	.540	.096	5.63
500	oI1600	1.136	.240	4.73
300	oI3200	2.217	.423	5.24
200	oI6400	4.420	.820	5.39
100	oI12800	8.880	1.660	5.35
			Average	4.76

Iterations	Statement	Scalar	Vector	Ratio
3000	oR100	.046	.027	1.74
3000	oR200	.072	.031	2.29
1000	oR400	.134	.048	2.79
500	oR800	.248	.078	3.18
500	oR1600	.558	.204	2.74
300	oR3200	1.083	.370	2.93
200	oR6400	2.160	.735	2.94
100	oR12800	4.420	1.450	3.05
			Average	2.71

Iterations	Statement	Scalar	Vector	Ratio
1000	oC100	.057	.035	1.63
750	oC200	.095	.044	2.15
500	oC400	.184	.088	2.09
400	oC800	.425	.225	1.89
300	oC1600	.807	.400	2.02
200	oC3200	1.630	.810	2.01
100	oC6400	3.360	1.670	2.01
50	oC12800	6.520	3.140	2.08
			Average	1.98

Iterations	Statement	Scalar	Vector	Ratio
3000	|I100	.045	.027	1.69
2000	|I200	.071	.032	2.20
1000	|I400	.121	.046	2.63
500	|I800	.232	.070	3.31
500	|I1600	.444	.120	3.70
250	|I3200	.944	.284	3.32
200	|I6400	1.855	.535	3.47
100	|I12800	3.710	1.070	3.47
			Average	2.97

Iterations	Statement	Scalar	Vector	Ratio
3000	|R100	.045	.027	1.68
2000	|R200	.071	.033	2.14
1000	|R400	.129	.049	2.63
500	|R800	.238	.080	2.98
500	|R1600	.532	.206	2.58
250	|R3200	1.028	.372	2.76
200	|R6400	2.060	.735	2.80
100	|R12800	4.200	1.440	2.92
			Average	2.56

Iterations	Statement	Scalar	Vector	Ratio
300	|C100	.400	.177	2.26
300	|C200	.777	.297	2.62
300	|C400	1.547	.570	2.71
200	|C800	3.080	1.115	2.76
200	|C1600	6.215	2.265	2.74
100	|C3200	12.430	4.460	2.79
100	|C6400	24.920	8.930	2.79
100	|C12800	49.760	17.800	2.80
			Average	2.68

Iterations	Statement	Scalar	Vector	Ratio
500	?B1000	.044	.026	1.69
400	?B2000	.065	.030	2.17
300	?B4000	.113	.037	3.09
200	?B8000	.205	.050	4.10
100	?B16000	.390	.080	4.88
50	?B32000	.820	.200	4.10
25	?B64000	1.640	.320	5.13
10	?B128000	3.100	.700	4.43
			Average	3.70

Dyadic Scalar Primitives

Iterations	Statement	Scalar	Vector	Ratio
1000	I100+I100D	.035	.034	1.03
1000	I200+I200D	.046	.038	1.21
1000	I400+I400D	.071	.052	1.37
500	I800+I800D	.116	.080	1.45
400	I1600+I1600D	.208	.130	1.60
300	I3200+I3200D	.470	.307	1.53
200	I6400+I6400D	.930	.585	1.59
100	I12800+I12800D	1.970	1.170	1.68
			Average	1.43

Iterations	Statement	Scalar	Vector	Ratio
1000	1234+I100	.035	.033	1.06
1000	1234+I200	.044	.036	1.22
1000	1234+I400	.068	.046	1.48
500	1234+I800	.110	.064	1.72
500	1234+I1600	.194	.100	1.94
300	1234+I3200	.440	.240	1.83
200	1234+I6400	.850	.435	1.95
100	1234+I12800	1.690	.870	1.94
			Average	1.64

Iterations	Statement	Scalar	Vector	Ratio
1000	R100+R100D	.039	.033	1.18
1000	R200+R200D	.049	.040	1.23
1000	R400+R400D	.082	.057	1.44
500	R800+R800D	.138	.084	1.64
400	R1600+R1600D	.340	.223	1.53
300	R3200+R3200D	.660	.417	1.58
200	R6400+R6400D	1.400	.835	1.68
100	R12800+R12800D	2.720	1.540	1.77
			Average	1.51

Iterations	Statement	Scalar	Vector	Ratio
1000	.1234+R100	.037	.032	1.16
1000	.1234+R200	.046	.036	1.28
1000	.1234+R400	.082	.051	1.61
500	.1234+R800	.138	.072	1.92
500	.1234+R1600	.332	.190	1.75
300	.1234+R3200	.627	.333	1.88
200	.1234+R6400	1.250	.650	1.92
100	.1234+R12800	2.570	1.260	2.04
			Average	1.69

Iterations	Statement	Scalar	Vector	Ratio
1000	C100+C100D	.048	.043	1.12
1000	C200+C200D	.070	.054	1.30
500	C400+C400D	.130	.098	1.33
200	C800+C800D	.325	.250	1.30
100	C1600+C1600D	.620	.450	1.38
50	C3200+C3200D	1.320	.960	1.38
50	C6400+C6400D	2.640	1.880	1.40
50	C12800+C12800D	5.140	3.620	1.42
			Average	1.33

Iterations	Statement	Scalar	Vector	Ratio
1000	1234J¯1235+C100	.047	.038	1.24
1000	1234J¯1235+C200	.068	.046	1.48
500	1234J¯1235+C400	.132	.084	1.57
200	1234J¯1235+C800	.320	.210	1.52
100	1234J¯1235+C1600	.590	.370	1.59
50	1234J¯1235+C3200	1.220	.760	1.61
50	1234J¯1235+C6400	2.440	1.520	1.61
50	1234J¯1235+C12800	4.720	2.900	1.63
			Average	1.53

Iterations	Statement	Scalar	Vector	Ratio
1000	I100-I100D	.035	.033	1.06
1000	I200-I200D	.046	.039	1.18
1000	I400-I400D	.067	.053	1.26
500	I800-I800D	.116	.080	1.45
400	I1600-I1600D	.205	.130	1.58
300	I3200-I3200D	.470	.307	1.53
200	I6400-I6400D	.925	.585	1.58
100	I12800-I12800D	1.980	1.160	1.71
			Average	1.42

Iterations	Statement	Scalar	Vector	Ratio
1000	1234-I100	.035	.032	1.09
1000	1234-I200	.044	.035	1.26
1000	1234-I400	.068	.046	1.48
500	1234-I800	.110	.064	1.72
500	1234-I1600	.194	.100	1.94
300	1234-I3200	.437	.240	1.82
200	1234-I6400	.835	.440	1.90
100	1234-I12800	1.690	.860	1.97
			Average	1.65

Iterations	Statement	Scalar	Vector	Ratio
1000	R100-R100D	.038	.033	1.15
1000	R200-R200D	.048	.039	1.23
1000	R400-R400D	.081	.056	1.45
500	R800-R800D	.138	.086	1.60
400	R1600-R1600D	.338	.218	1.55
300	R3200-R3200D	.663	.407	1.63
200	R6400-R6400D	1.425	.815	1.75
100	R12800-R12800D	2.730	1.520	1.80
			Average	1.52

Iterations	Statement	Scalar	Vector	Ratio
1000	.1234-R100	.037	.032	1.16
1000	.1234-R200	.049	.035	1.40
1000	.1234-R400	.083	.049	1.69
500	.1234-R800	.140	.072	1.94
500	.1234-R1600	.334	.186	1.80
300	.1234-R3200	.627	.327	1.92
200	.1234-R6400	1.265	.645	1.96
100	.1234-R12800	2.590	1.230	2.11
			Average	1.75

Iterations	Statement	Scalar	Vector	Ratio
1000	C100-C100D	.046	.043	1.07
1000	C200-C200D	.068	.052	1.31
500	C400-C400D	.132	.098	1.35
200	C800-C800D	.335	.255	1.31
100	C1600-C1600D	.620	.470	1.32
50	C3200-C3200D	1.380	1.000	1.38
50	C6400-C6400D	2.640	1.920	1.38
50	C12800-C12800D	5.220	3.760	1.39
			Average	1.31

Iterations	Statement	Scalar	Vector	Ratio
1000	1234J¯1235-C100	.046	.038	1.21
1000	1234J¯1235-C200	.069	.045	1.53
500	1234J¯1235-C400	.132	.084	1.57
200	1234J¯1235-C800	.315	.210	1.50
100	1234J¯1235-C1600	.580	.360	1.61
50	1234J¯1235-C3200	1.180	.740	1.59
50	1234J¯1235-C6400	2.400	1.460	1.64
50	1234J¯1235-C12800	4.660	2.800	1.66
			Average	1.54

Iterations	Statement	Scalar	Vector	Ratio
500	B800M×B800MM	.030	.032	.94
400	B1600M×B1600MM	.033	.035	.93
300	B3200M×B3200MM	.040	.037	1.09
200	B6400M×B6400MM	.055	.040	1.38
100	B12800M×B12800MM	.070	.050	1.40
			Average	1.15

Iterations	Statement	Scalar	Vector	Ratio
1000	I100×I100D	.045	.044	1.02
1000	I200×I200D	.066	.056	1.18
1000	I400×I400D	.108	.087	1.24
500	I800×I800D	.192	.146	1.32
400	I1600×I1600D	.358	.260	1.38
300	I3200×I3200D	.767	.567	1.35
200	I6400×I6400D	1.535	1.110	1.38
100	I12800×I12800D	3.150	2.220	1.42
			Average	1.29

Iterations	Statement	Scalar	Vector	Ratio
1000	1234×I100	.045	.042	1.07
1000	1234×I200	.065	.052	1.25
1000	1234×I400	.107	.079	1.35
500	1234×I800	.190	.132	1.44
500	1234×I1600	.352	.232	1.52
300	1234×I3200	.760	.503	1.51
200	1234×I6400	1.485	.960	1.55
100	1234×I12800	2.940	1.910	1.54
			Average	1.40

Iterations	Statement	Scalar	Vector	Ratio
1000	R100×R100D	.040	.033	1.21
1000	R200×R200D	.053	.039	1.36
1000	R400×R400D	.088	.056	1.57
500	R800×R800D	.152	.084	1.81
400	R1600×R1600D	.365	.230	1.59
300	R3200×R3200D	.707	.410	1.72
200	R6400×R6400D	1.500	.815	1.84
100	R12800×R12800D	2.910	1.500	1.94
			Average	1.63

Iterations	Statement	Scalar	Vector	Ratio
1000	.1234×R100	.040	.033	1.21
1000	.1234×R200	.057	.035	1.63
1000	.1234×R400	.092	.049	1.88
500	.1234×R800	.158	.072	2.19
500	.1234×R1600	.372	.188	1.98
300	.1234×R3200	.693	.330	2.10
200	.1234×R6400	1.405	.650	2.16
100	.1234×R12800	2.880	1.260	2.29
			Average	1.93

Iterations	Statement	Scalar	Vector	Ratio
1000	C100×C100D	.080	.062	1.29
1000	C200×C200D	.130	.084	1.55
500	C400×C400D	.254	.154	1.65
200	C800×C800D	.565	.350	1.61
100	C1600×C1600D	1.110	.670	1.66
50	C3200×C3200D	2.300	1.400	1.64
50	C6400×C6400D	4.520	2.680	1.69
50	C12800×C12800D	8.840	5.260	1.68
			Average	1.60

Iterations	Statement	Scalar	Vector	Ratio
1000	1234J¯1235×C100	.078	.058	1.34
1000	1234J¯1235×C200	.132	.078	1.69
500	1234J¯1235×C400	.252	.142	1.77
200	1234J¯1235×C800	.555	.325	1.71
100	1234J¯1235×C1600	1.060	.590	1.80
50	1234J¯1235×C3200	2.120	1.180	1.80
50	1234J¯1235×C6400	4.260	2.340	1.82
50	1234J¯1235×C12800	8.400	4.540	1.85
			Average	1.72

Iterations	Statement	Scalar	Vector	Ratio
1000	1234+I100	.161	.070	2.30
1000	1234+I200	.276	.088	3.14
1000	1234+I400	.505	.135	3.74
500	1234+I800	.962	.220	4.37
500	1234+I1600	1.962	.468	4.19
300	1234+I3200	3.867	.850	4.55
200	1234+I6400	7.665	1.650	4.65
100	1234+I12800	15.320	3.260	4.70
			Average	3.95

Iterations	Statement	Scalar	Vector	Ratio
1000	R100+R100D	.091	.050	1.82
1000	R200+R200D	.154	.066	2.33
1000	R400+R400D	.282	.110	2.56
500	R800+R800D	.542	.194	2.79
400	R1600+R1600D	1.148	.435	2.64
300	R3200+R3200D	2.270	.837	2.71
200	R6400+R6400D	4.560	1.660	2.75
100	R12800+R12800D	9.010	3.210	2.81
			Average	2.55

Iterations	Statement	Scalar	Vector	Ratio
1000	.1234+R100	.089	.047	1.89
1000	.1234+R200	.154	.064	2.41
1000	.1234+R400	.283	.105	2.70
500	.1234+R800	.540	.184	2.93
500	.1234+R1600	1.136	.410	2.77
300	.1234+R3200	2.217	.767	2.89
200	.1234+R6400	4.450	1.525	2.92
100	.1234+R12800	8.930	3.030	2.95
			Average	2.68

Iterations	Statement	Scalar	Vector	Ratio
1000	C100+C100D	.389	.124	3.14
1000	C200+C200D	.751	.202	3.72
500	C400+C400D	1.484	.392	3.79
200	C800+C800D	3.020	.825	3.66
100	C1600+C1600D	6.020	1.610	3.74
50	C3200+C3200D	12.140	3.260	3.72
50	C6400+C6400D	24.220	6.420	3.77
50	C12800+C12800D	48.260	12.700	3.80
			Average	3.67

Iterations	Statement	Scalar	Vector	Ratio
1000	1234J¯1235+C100	.394	.121	3.26
1000	1234J¯1235+C200	.760	.197	3.86
500	1234J¯1235+C400	1.488	.380	3.92
200	1234J¯1235+C800	3.035	.800	3.79
100	1234J¯1235+C1600	6.040	1.550	3.90
50	1234J¯1235+C3200	12.040	3.060	3.93
50	1234J¯1235+C6400	24.140	6.140	3.93
50	1234J¯1235+C12800	48.100	12.060	3.99
			Average	3.82

Iterations	Statement	Scalar	Vector	Ratio
500	B800MLB800MM	.032	.034	.94
400	B1600MLB1600MM	.035	.033	1.08
300	B3200MLB3200MM	.040	.037	1.09
200	B6400MLB6400MM	.050	.040	1.25
100	B12800MLB12800MM	.080	.060	1.33
			Average	1.14

Iterations	Statement	Scalar	Vector	Ratio
1000	I100LI100D	.042	.036	1.17
1000	I200LI200D	.062	.042	1.48
1000	I400LI400D	.097	.059	1.64
500	I800LI800D	.172	.092	1.87
400	I1600LI1600D	.320	.155	2.06
300	I3200LI3200D	.710	.370	1.92
200	I6400LI6400D	1.390	.695	2.00
100	I12800LI12800D	2.870	1.400	2.05
			Average	1.77

Iterations	Statement	Scalar	Vector	Ratio
1000	R100LR100D	.045	.035	1.29
1000	R200LR200D	.066	.041	1.61
1000	R400LR400D	.108	.062	1.74
500	R800LR800D	.196	.100	1.96
400	R1600LR1600D	.458	.253	1.81
300	R3200LR3200D	.897	.473	1.89
200	R6400LR6400D	1.905	.945	2.02
100	R12800LR12800D	3.720	1.760	2.11
			Average	1.80

Iterations	Statement	Scalar	Vector	Ratio
500	B800M⌈B800MM	.030	.032	.94
400	B1600M⌈B1600MM	.033	.035	.93
300	B3200M⌈B3200MM	.040	.037	1.09
200	B6400M⌈B6400MM	.055	.040	1.38
100	B12800M⌈B12800MM	.070	.050	1.40
			Average	1.15

Iterations	Statement	Scalar	Vector	Ratio
1000	I100⌈I100D	.042	.036	1.17
1000	I200⌈I200D	.062	.042	1.48
1000	I400⌈I400D	.099	.060	1.65
500	I800⌈I800D	.172	.092	1.87
400	I1600⌈I1600D	.328	.158	2.08
300	I3200⌈I3200D	.720	.367	1.96
200	I6400⌈I6400D	1.405	.700	2.01
100	I12800⌈I12800D	2.920	1.410	2.07
			Average	1.79

Iterations	Statement	Scalar	Vector	Ratio
1000	R100⌈R100D	.044	.036	1.22
1000	R200⌈R200D	.064	.042	1.52
1000	R400⌈R400D	.110	.063	1.75
500	R800⌈R800D	.196	.100	1.96
400	R1600⌈R1600D	.460	.255	1.80
300	R3200⌈R3200D	.900	.473	1.90
200	R6400⌈R6400D	1.910	.945	2.02
100	R12800⌈R12800D	3.730	1.760	2.12
			Average	1.79

Iterations	Statement	Scalar	Vector	Ratio
500	B800M*B800MM	.036	.034	1.06
400	B1600M*B1600MM	.043	.035	1.21
300	B3200M*B3200MM	.053	.037	1.45
200	B6400M*B6400MM	.085	.040	2.13
100	B12800M*B12800MM	.130	.060	2.17
			Average	1.60

Iterations	Statement	Scalar	Vector	Ratio
50	R100*PR100	.500	.200	2.50
50	R200*PR200	.980	.360	2.72
50	R400*PR400	1.920	.680	2.82
50	R800*PR800	3.820	1.320	2.89
50	R1600*PR1600	7.720	2.660	2.90
50	R3200*PR3200	15.400	5.240	2.94
50	R6400*PR6400	30.840	10.420	2.96
50	R12800*PR12800	61.580	20.700	2.97
			Average	2.84

Iterations	Statement	Scalar	Vector	Ratio
50	I100*.5	.520	.220	2.36
50	I200*.5	1.020	.340	3.00
25	I400*.5	2.000	.680	2.94
25	I800*.5	4.000	1.360	2.94
15	I1600*.5	8.000	3.800	2.11
15	I3200*.5	16.067	5.267	3.05
10	I6400*.5	32.000	11.500	2.78
5	I12800*.5	64.000	21.800	2.94
			Average	2.77

Iterations	Statement	Scalar	Vector	Ratio
50	R100*.5	.300	.100	3.00
50	R200*.5	.560	.160	3.50
25	R400*.5	1.080	.320	3.38
25	R800*.5	2.160	.600	3.60
15	R1600*.5	4.333	1.200	3.61
15	R3200*.5	8.667	2.400	3.61
10	R6400*.5	17.400	4.700	3.70
5	R12800*.5	34.600	9.400	3.68
			Average	3.51

Iterations	Statement	Scalar	Vector	Ratio
1000	I100*2	.042	.033	1.27
1000	I200*2	.061	.040	1.53
1000	I400*2	.103	.061	1.69
500	I800*2	.184	.100	1.84
400	I1600*2	.348	.175	1.99
300	I3200*2	.757	.400	1.89
200	I6400*2	1.450	.740	1.96
100	I12800*2	2.930	1.510	1.94
			Average	1.76

Iterations	Statement	Scalar	Vector	Ratio
1000	R100*2	.032	.030	1.07
1000	R200*2	.044	.034	1.29
1000	R400*2	.075	.050	1.50
500	R800*2	.130	.080	1.63
400	R1600*2	.328	.218	1.51
300	R3200*2	.590	.370	1.59
200	R6400*2	1.195	.740	1.61
100	R12800*2	2.450	1.450	1.69
			Average	1.49

Iterations	Statement	Scalar	Vector	Ratio
500	10●I100	.330	.122	2.70
500	10●I200	.636	.198	3.21
500	10●I400	1.254	.376	3.34
100	10●I800	2.490	.770	3.23
100	10●I1600	5.030	1.480	3.40
100	10●I3200	10.030	2.860	3.51
100	10●I6400	20.030	5.690	3.52
100	10●I12800	40.000	11.340	3.53
			Average	3.30

Iterations	Statement	Scalar	Vector	Ratio
500	10●R100	.296	.120	2.47
500	10●R200	.572	.194	2.95
500	10●R400	1.128	.366	3.08
100	10●R800	2.230	.710	3.14
100	10●R1600	4.530	1.440	3.15
100	10●R3200	9.030	2.820	3.20
100	10●R6400	18.050	5.600	3.22
100	10●R12800	36.100	11.130	3.24
			Average	3.06

Iterations	Statement	Scalar	Vector	Ratio
500	B800M=B800MM	.030	.034	.88
400	B1600M=B1600MM	.035	.035	1.00
300	B3200M=B3200MM	.043	.040	1.08
200	B6400M=B6400MM	.060	.045	1.33
200	B12800M=B12800MM	.095	.060	1.58
			Average	1.18

Iterations	Statement	Scalar	Vector	Ratio
1000	I100=I100D	.037	.030	1.23
500	I200=I200D	.052	.036	1.44
500	I400=I400D	.080	.046	1.74
200	I800=I800D	.135	.065	2.08
200	I1600=I1600D	.240	.095	2.53
200	I3200=I3200D	.465	.160	2.91
200	I6400=I6400D	.890	.290	3.07
200	I12800=I12800D	1.830	.595	3.08
			Average	2.26

Iterations	Statement	Scalar	Vector	Ratio
1000	R100=R100D	.132	.060	2.20
500	R200=R200D	.242	.088	2.75
500	R400=R400D	.460	.146	3.15
200	R800=R800D	.895	.260	3.44
200	R1600=R1600D	1.765	.485	3.64
200	R3200=R3200D	3.505	.930	3.77
200	R6400=R6400D	7.040	1.855	3.80
200	R12800=R12800D	14.400	3.755	3.83
			Average	3.32

Iterations	Statement	Scalar	Vector	Ratio
500	B800M≠B800MM	.030	.032	.94
400	B1600M≠B1600MM	.033	.033	1.00
300	B3200M≠B3200MM	.040	.033	1.20
200	B6400M≠B6400MM	.050	.040	1.25
200	B12800M≠B12800MM	.075	.055	1.36
			Average	1.15

Iterations	Statement	Scalar	Vector	Ratio
1000	I100≠I100D	.037	.030	1.23
500	I200≠I200D	.054	.036	1.50
500	I400≠I400D	.078	.046	1.70
200	I800≠I800D	.135	.060	2.25
200	I1600≠I1600D	.240	.095	2.53
200	I3200≠I3200D	.460	.160	2.88
200	I6400≠I6400D	.885	.295	3.00
200	I12800≠I12800D	1.835	.595	3.08
			Average	2.27

Iterations	Statement	Scalar	Vector	Ratio
1000	R100≠R100D	.121	.061	1.98
500	R200≠R200D	.222	.090	2.47
500	R400≠R400D	.416	.146	2.85
200	R800≠R800D	.810	.265	3.06
200	R1600≠R1600D	1.595	.490	3.26
200	R3200≠R3200D	3.165	.935	3.39
200	R6400≠R6400D	6.375	1.870	3.41
200	R12800≠R12800D	13.035	3.805	3.43
			Average	2.98

Iterations	Statement	Scalar	Vector	Ratio
500	B800M<B800MM	.034	.032	1.06
400	B1600M<B1600MM	.040	.035	1.14
300	B3200M<B3200MM	.053	.037	1.45
200	B6400M<B6400MM	.085	.040	2.13
200	B12800M<B12800MM	.130	.055	2.36
			Average	1.63

Iterations	Statement	Scalar	Vector	Ratio
1000	I100<I100D	.040	.030	1.33
500	I200<I200D	.058	.038	1.53
500	I400<I400D	.092	.044	2.09
200	I800<I800D	.155	.060	2.58
200	I1600<I1600D	.285	.095	3.00
200	I3200<I3200D	.540	.160	3.38
200	I6400<I6400D	1.050	.290	3.62
200	I12800<I12800D	2.155	.600	3.59
			Average	2.64

Iterations	Statement	Scalar	Vector	Ratio
1000	R100<R100D	.082	.061	1.34
500	R200<R200D	.150	.090	1.67
500	R400<R400D	.290	.148	1.96
200	R800<R800D	.530	.260	2.04
200	R1600<R1600D	1.065	.490	2.17
200	R3200<R3200D	2.035	.935	2.18
200	R6400<R6400D	4.080	1.865	2.19
200	R12800<R12800D	8.540	3.755	2.27
			Average	1.98

Iterations	Statement	Scalar	Vector	Ratio
500	B800M>B800MM	.032	.032	1.00
400	B1600M>B1600MM	.040	.033	1.23
300	B3200M>B3200MM	.053	.033	1.60
200	B6400M>B6400MM	.080	.040	2.00
200	B12800M>B12800MM	.135	.055	2.45
			Average	1.66

Iterations	Statement	Scalar	Vector	Ratio
1000	I100>I100D	.040	.030	1.33
500	I200>I200D	.058	.036	1.61
500	I400>I400D	.088	.044	2.00
200	I800>I800D	.155	.065	2.38
200	I1600>I1600D	.280	.095	2.95
200	I3200>I3200D	.545	.160	3.41
200	I6400>I6400D	1.050	.290	3.62
200	I12800>I12800D	2.170	.600	3.62
			Average	2.62

Iterations	Statement	Scalar	Vector	Ratio
1000	R100>R100D	.090	.061	1.48
500	R200>R200D	.156	.090	1.73
500	R400>R400D	.266	.148	1.80
200	R800>R800D	.530	.265	2.00
200	R1600>R1600D	1.020	.490	2.08
200	R3200>R3200D	2.055	.935	2.20
200	R6400>R6400D	4.200	1.870	2.25
200	R12800>R12800D	8.575	3.760	2.28
			Average	1.98

Iterations	Statement	Scalar	Vector	Ratio
500	B800M≤B800MM	.032	.032	1.00
400	B1600M≤B1600MM	.040	.033	1.23
300	B3200M≤B3200MM	.053	.037	1.45
200	B6400M≤B6400MM	.085	.040	2.13
200	B12800M≤B12800MM	.135	.055	2.45
			Average	1.65

Iterations	Statement	Scalar	Vector	Ratio
1000	I100≤I100D	.040	.031	1.29
500	I200≤I200D	.058	.036	1.61
500	I400≤I400D	.088	.044	2.00
200	I800≤I800D	.150	.060	2.50
200	I1600≤I1600D	.280	.095	2.95
200	I3200≤I3200D	.540	.160	3.38
200	I6400≤I6400D	1.050	.295	3.56
200	I12800≤I12800D	2.175	.600	3.63
			Average	2.61

Iterations	Statement	Scalar	Vector	Ratio
1000	R100≤R100D	.095	.061	1.56
500	R200≤R200D	.166	.088	1.89
500	R400≤R400D	.280	.144	1.94
200	R800≤R800D	.570	.260	2.19
200	R1600≤R1600D	1.085	.485	2.24
200	R3200≤R3200D	2.200	.925	2.38
200	R6400≤R6400D	4.500	1.855	2.43
200	R12800≤R12800D	9.195	3.755	2.45
			Average	2.13

Iterations	Statement	Scalar	Vector	Ratio
500	B800M≥B800MM	.034	.030	1.13
400	B1600M≥B1600MM	.040	.033	1.23
300	B3200M≥B3200MM	.053	.037	1.45
200	B6400M≥B6400MM	.080	.040	2.00
200	B12800M≥B12800MM	.135	.055	2.45
			Average	1.65

Iterations	Statement	Scalar	Vector	Ratio
1000	I100≥I100D	.039	.030	1.30
500	I200≥I200D	.058	.036	1.61
500	I400≥I400D	.090	.044	2.05
200	I800≥I800D	.155	.060	2.58
200	I1600≥I1600D	.285	.095	3.00
200	I3200≥I3200D	.540	.160	3.38
200	I6400≥I6400D	1.050	.290	3.62
200	I12800≥I12800D	2.185	.615	3.55
			Average	2.64

Iterations	Statement	Scalar	Vector	Ratio
1000	R100≥R100D	.087	.061	1.43
500	R200≥R200D	.158	.088	1.80
500	R400≥R400D	.310	.146	2.12
200	R800≥R800D	.565	.260	2.17
200	R1600≥R1600D	1.140	.480	2.38
200	R3200≥R3200D	2.180	.925	2.36
200	R6400≥R6400D	4.385	1.865	2.35
200	R12800≥R12800D	9.170	3.780	2.43
			Average	2.13

Iterations	Statement	Scalar	Vector	Ratio
100	B800M|B800MM	.040	.030	1.33
100	B1600M|B1600MM	.040	.040	1.00
50	B3200M|B3200MM	.040	.040	1.00
25	B6400M|B6400MM	.080	.040	2.00
25	B12800M|B12800MM	.160	.080	2.00
			Average	1.47

Iterations	Statement	Scalar	Vector	Ratio
500	B800M!B800MM	.032	.030	1.07
400	B1600M!B1600MM	.040	.033	1.23
300	B3200M!B3200MM	.053	.033	1.60
200	B6400M!B6400MM	.080	.040	2.00
100	B12800M!B12800MM	.130	.050	2.60
			Average	1.70

Iterations	Statement	Scalar	Vector	Ratio
500	B800M∧B800MM	.032	.032	1.00
400	B1600M∧B1600MM	.033	.030	1.08
300	B3200M∧B3200MM	.040	.037	1.09
200	B6400M∧B6400MM	.050	.040	1.25
100	B12800M∧B12800MM	.080	.060	1.33
			Average	1.15

Iterations	Statement	Scalar	Vector	Ratio
500	B800M∨B800MM	.030	.032	.94
400	B1600M∨B1600MM	.030	.035	.86
300	B3200M∨B3200MM	.037	.033	1.10
200	B6400M∨B6400MM	.050	.040	1.25
100	B12800M∨B12800MM	.070	.050	1.40
			Average	1.11

Iterations	Statement	Scalar	Vector	Ratio
500	B800M⍲B800MM	.030	.030	1.00
400	B1600M⍲B1600MM	.035	.033	1.08
300	B3200M⍲B3200MM	.043	.037	1.18
200	B6400M⍲B6400MM	.055	.045	1.22
100	B12800M⍲B12800MM	.090	.060	1.50
			Average	1.20

Iterations	Statement	Scalar	Vector	Ratio
500	B800M⍱B800MM	.032	.034	.94
400	B1600M⍱B1600MM	.033	.033	1.00
300	B3200M⍱B3200MM	.043	.037	1.18
200	B6400M⍱B6400MM	.060	.045	1.33
100	B12800M⍱B12800MM	.090	.060	1.50
			Average	1.19

Circular Functions

Iterations	Statement	Scalar	Vector	Ratio
500	0○B100M	.058	.038	1.53
500	0○B200M	.088	.040	2.20
500	0○B400M	.150	.058	2.59
200	0○B800M	.265	.085	3.12
200	0○B1600M	.610	.210	2.90
50	0○B3200M	1.120	.360	3.11
50	0○B6400M	2.280	.700	3.26
50	0○B12800M	4.460	1.300	3.43
			Average	2.77

Iterations	Statement	Scalar	Vector	Ratio
1000	0○R100	.361	.122	2.96
1000	0○R200	.704	.199	3.54
1000	0○R400	1.394	.379	3.68
500	0○R800	2.758	.724	3.81
400	0○R1600	5.645	1.528	3.70
300	0○R3200	11.157	2.923	3.82
200	0○R6400	22.220	5.775	3.85
100	0○R12800	44.280	11.390	3.89
			Average	3.65

Iterations	Statement	Scalar	Vector	Ratio
500	1○B100M	.052	.034	1.53
500	1○B200M	.082	.040	2.05
500	1○B400M	.146	.058	2.52
200	1○B800M	.270	.085	3.18
200	1○B1600M	.595	.205	2.90
50	1○B3200M	1.140	.340	3.35
50	1○B6400M	2.260	.640	3.53
50	1○B12800M	4.480	1.220	3.67
			Average	2.84

Iterations	Statement	Scalar	Vector	Ratio
1000	1○I100	.299	.123	2.43
1000	1○I200	.574	.198	2.90
1000	1○I400	1.133	.415	2.73
500	1○I800	2.236	.754	2.97
400	1○I1600	4.585	1.538	2.98
300	1○I3200	9.067	3.030	2.99
200	1○I6400	18.045	6.025	3.00
100	1○I12800	35.880	11.680	3.07
			Average	2.88

Iterations	Statement	Scalar	Vector	Ratio
1000	1○R100	.173	.118	1.47
1000	1○R200	.368	.213	1.73
1000	1○R400	.859	.400	2.15
500	1○R800	1.826	.766	2.38
400	1○R1600	3.880	1.573	2.47
300	1○R3200	7.697	3.043	2.53
200	1○R6400	15.505	6.060	2.56
100	1○R12800	30.920	11.990	2.58
			Average	2.23

Iterations	Statement	Scalar	Vector	Ratio
500	2oB100M	.054	.034	1.59
500	2oB200M	.082	.038	2.16
500	2oB400M	.148	.058	2.55
200	2oB800M	.270	.085	3.18
200	2oB1600M	.605	.215	2.81
50	2oB3200M	1.140	.400	2.85
50	2oB6400M	2.240	.660	3.39
50	2oB12800M	4.400	1.360	3.24
			Average	2.72

Iterations	Statement	Scalar	Vector	Ratio
1000	2oI100	.294	.133	2.21
1000	2oI200	.567	.216	2.63
1000	2oI400	1.118	.417	2.68
500	2oI800	2.206	.770	2.86
400	2oI1600	4.508	1.645	2.74
300	2oI3200	8.923	3.123	2.86
200	2oI6400	17.770	6.135	2.90
100	2oI12800	35.260	12.230	2.88
			Average	2.72

Iterations	Statement	Scalar	Vector	Ratio
1000	2oR100	.201	.138	1.46
1000	2oR200	.385	.205	1.88
1000	2oR400	.856	.390	2.19
500	2oR800	1.788	.770	2.32
400	2oR1600	3.790	1.613	2.35
300	2oR3200	7.613	3.137	2.43
200	2oR6400	15.235	6.190	2.46
100	2oR12800	30.370	12.270	2.48
			Average	2.20

Iterations	Statement	Scalar	Vector	Ratio
500	3oB100M	.054	.034	1.59
500	3oB200M	.082	.040	2.05
500	3oB400M	.148	.058	2.55
200	3oB800M	.270	.085	3.18
200	3oB1600M	.615	.230	2.67
50	3oB3200M	1.160	.380	3.05
50	3oB6400M	2.260	.640	3.53
50	3oB12800M	4.400	1.260	3.49
			Average	2.76

Iterations	Statement	Scalar	Vector	Ratio
1000	3oI100	.419	.294	1.43
1000	3oI200	.825	.548	1.51
1000	3oI400	1.610	1.043	1.54
500	3oI800	3.204	2.018	1.59
400	3oI1600	6.493	4.100	1.58
300	3oI3200	12.887	8.123	1.59
200	3oI6400	25.625	16.075	1.59
100	3oI12800	51.120	31.880	1.60
			Average	1.55

Iterations	Statement	Scalar	Vector	Ratio
1000	3oR100	.287	.229	1.25
1000	3oR200	.620	.374	1.66
1000	3oR400	1.296	.726	1.79
500	3oR800	2.652	1.398	1.90
400	3oR1600	5.503	2.913	1.89
300	3oR3200	11.023	5.697	1.94
200	3oR6400	22.020	11.260	1.96
100	3oR12800	43.930	22.380	1.96
			Average	1.79

Iterations	Statement	Scalar	Vector	Ratio
500	4oB100M	.056	.038	1.47
500	4oB200M	.084	.044	1.91
500	4oB400M	.152	.062	2.45
200	4oB800M	.280	.085	3.29
200	4oB1600M	.620	.230	2.70
50	4oB3200M	1.160	.400	2.90
50	4oB6400M	2.260	.680	3.32
50	4oB12800M	4.440	1.300	3.42
			Average	2.68

Iterations	Statement	Scalar	Vector	Ratio
1000	4oI100	.383	.123	3.11
1000	4oI200	.741	.199	3.72
1000	4oI400	1.470	.375	3.92
500	4oI800	2.902	.724	4.01
400	4oI1600	5.923	1.503	3.94
300	4oI3200	11.747	2.950	3.98
200	4oI6400	23.345	5.765	4.05
100	4oI12800	46.610	11.400	4.09
			Average	3.85

Iterations	Statement	Scalar	Vector	Ratio
1000	4oR100	.348	.122	2.85
1000	4oR200	.676	.192	3.52
1000	4oR400	1.329	.368	3.61
500	4oR800	2.640	.708	3.73
400	4oR1600	5.385	1.488	3.62
300	4oR3200	10.690	2.890	3.70
200	4oR6400	21.310	5.645	3.78
100	4oR12800	42.480	11.230	3.78
			Average	3.57

Iterations	Statement	Scalar	Vector	Ratio
500	5oB100M	.054	.036	1.50
500	5oB200M	.084	.040	2.10
500	5oB400M	.150	.060	2.50
200	5oB800M	.270	.085	3.18
200	5oB1600M	.595	.220	2.70
50	5oB3200M	1.140	.400	2.85
50	5oB6400M	2.280	.680	3.35
50	5oB12800M	4.380	1.240	3.53
			Average	2.71

Iterations	Statement	Scalar	Vector	Ratio
500	6oB100M	.058	.040	1.45
500	6oB200M	.088	.044	2.00
500	6oB400M	.152	.062	2.45
200	6oB800M	.275	.090	3.06
200	6oB1600M	.615	.225	2.73
50	6oB3200M	1.180	.360	3.28
50	6oB6400M	2.300	.740	3.11
50	6oB12800M	4.360	1.320	3.30
			Average	2.67

Iterations	Statement	Scalar	Vector	Ratio
500	7oB100M	.054	.036	1.50
500	7oB200M	.084	.042	2.00
500	7oB400M	.150	.058	2.59
200	7oB800M	.275	.085	3.24
200	7oB1600M	.610	.205	2.98
50	7oB3200M	1.120	.380	2.95
50	7oB6400M	2.300	.660	3.48
50	7oB12800M	4.360	1.260	3.46
			Average	2.77

Iterations	Statement	Scalar	Vector	Ratio
500	8oB100M	.066	.050	1.32
500	8oB200M	.096	.062	1.55
500	8oB400M	.182	.102	1.78
200	8oB800M	.410	.270	1.52
200	8oB1600M	.775	.430	1.80
50	8oB3200M	1.440	.760	1.89
50	8oB6400M	2.940	1.400	2.10
50	8oB12800M	5.640	2.740	2.06
			Average	1.75

Iterations	Statement	Scalar	Vector	Ratio
2000	9oC100	.056	.033	1.72
2000	9oC200	.090	.038	2.37
500	9oC400	.168	.060	2.80
300	9oC800	.313	.093	3.36
200	9oC1600	.720	.255	2.82
100	9oC3200	1.450	.520	2.79
100	9oC6400	2.880	1.030	2.80
50	9oC12800	5.600	1.900	2.95
			Average	2.70

Iterations	Statement	Scalar	Vector	Ratio
1000	10oI100	.050	.030	1.67
1000	10oI200	.074	.034	2.18
1000	10oI400	.133	.050	2.66
500	10oI800	.242	.076	3.18
400	10oI1600	.463	.125	3.70
300	10oI3200	1.000	.323	3.09
200	10oI6400	1.970	.595	3.31
100	10oI12800	3.920	1.140	3.44
			Average	2.90

Iterations	Statement	Scalar	Vector	Ratio
1000	10oR100	.048	.031	1.55
1000	10oR200	.074	.034	2.18
1000	10oR400	.136	.053	2.57
500	10oR800	.250	.084	2.98
400	10oR1600	.588	.243	2.42
300	10oR3200	1.130	.423	2.67
200	10oR6400	2.205	.845	2.61
100	10oR12800	4.380	1.580	2.77
			Average	2.47

Iterations	Statement	Scalar	Vector	Ratio
1000	10oC100	.409	.181	2.26
1000	10oC200	.800	.306	2.61
1000	10oC400	1.585	.586	2.70
500	10oC800	3.136	1.152	2.72
400	10oC1600	6.398	2.380	2.69
300	10oC3200	12.700	4.630	2.74
200	10oC6400	25.265	9.130	2.77
100	10oC12800	50.420	18.150	2.78
			Average	2.66

Iterations	Statement	Scalar	Vector	Ratio
1000	11oI100	.049	.029	1.69
1000	11oI200	.074	.032	2.31
1000	11oI400	.131	.045	2.91
500	11oI800	.244	.066	3.70
400	11oI1600	.455	.108	4.23
300	11oI3200	1.000	.283	3.53
200	11oI6400	1.965	.535	3.67
100	11oI12800	3.900	.970	4.02
			Average	3.26

Iterations	Statement	Scalar	Vector	Ratio
1000	11oR100	.049	.029	1.69
1000	11oR200	.075	.032	2.34
1000	11oR400	.136	.049	2.78
500	11oR800	.252	.074	3.41
400	11oR1600	.578	.218	2.66
300	11oR3200	1.120	.367	3.05
200	11oR6400	2.210	.740	2.99
100	11oR12800	4.330	1.430	3.03
			Average	2.74

Iterations	Statement	Scalar	Vector	Ratio
2000	11oC100	.052	.037	1.39
2000	11oC200	.081	.046	1.75
500	11oC400	.164	.092	1.78
300	11oC800	.407	.250	1.63
200	11oC1600	.780	.465	1.68
100	11oC3200	1.520	.930	1.63
100	11oC6400	3.070	1.750	1.75
50	11oC12800	5.940	3.340	1.78
			Average	1.67

Iterations	Statement	Scalar	Vector	Ratio
500	12oB100M	.050	.032	1.56
500	12oB200M	.080	.036	2.22
500	12oB400M	.146	.054	2.70
200	12oB800M	.270	.085	3.18
200	12oB1600M	.595	.210	2.83
50	12oB3200M	1.160	.360	3.22
50	12oB6400M	2.280	.660	3.45
50	12oB12800M	4.400	1.240	3.55
			Average	2.84

Iterations	Statement	Scalar	Vector	Ratio
1000	12oI100	.089	.036	2.47
1000	12oI200	.157	.044	3.57
1000	12oI400	.302	.071	4.25
500	12oI800	.582	.118	4.93
400	12oI1600	1.253	.310	4.04
300	12oI3200	2.430	.550	4.42
200	12oI6400	4.785	1.065	4.49
100	12oI12800	9.540	2.010	4.75
			Average	4.12

Iterations	Statement	Scalar	Vector	Ratio
1000	12oR100	.055	.032	1.72
1000	12oR200	.087	.040	2.18
1000	12oR400	.160	.063	2.54
500	12oR800	.296	.100	2.96
400	12oR1600	.678	.275	2.46
300	12oR3200	1.313	.507	2.59
200	12oR6400	2.630	.970	2.71
100	12oR12800	5.200	1.840	2.83
			Average	2.50

Iterations	Statement	Scalar	Vector	Ratio
500	¯1oB100M	.052	.034	1.53
500	¯1oB200M	.080	.038	2.11
500	¯1oB400M	.146	.056	2.61
200	¯1oB800M	.275	.085	3.24
200	¯1oB1600M	.595	.205	2.90
50	¯1oB3200M	1.180	.360	3.28
50	¯1oB6400M	2.220	.660	3.36
50	¯1oB12800M	4.340	1.340	3.24
			Average	2.78

Iterations	Statement	Scalar	Vector	Ratio
500	¯2oB100M	.052	.034	1.53
500	¯2oB200M	.080	.040	2.00
500	¯2oB400M	.146	.056	2.61
200	¯2oB800M	.265	.085	3.12
200	¯2oB1600M	.605	.225	2.69
50	¯2oB3200M	1.180	.380	3.11
50	¯2oB6400M	2.220	.720	3.08
50	¯2oB12800M	4.380	1.280	3.42
			Average	2.69

Iterations	Statement	Scalar	Vector	Ratio
500	¯3○B100M	.054	.036	1.50
500	¯3○B200M	.080	.040	2.00
500	¯3○B400M	.146	.056	2.61
200	¯3○B800M	.275	.080	3.44
200	¯3○B1600M	.600	.205	2.93
50	¯3○B3200M	1.160	.360	3.22
50	¯3○B6400M	2.240	.700	3.20
50	¯3○B12800M	4.440	1.260	3.52
			Average	2.80

Iterations	Statement	Scalar	Vector	Ratio
500	¯3○I100	.252	.208	1.21
500	¯3○I200	.482	.382	1.26
500	¯3○I400	.950	.746	1.27
200	¯3○I800	1.875	1.485	1.26
200	¯3○I1600	3.845	3.030	1.27
50	¯3○I3200	7.680	5.920	1.30
50	¯3○I6400	15.200	11.760	1.29
50	¯3○I12800	30.220	23.400	1.29
			Average	1.27

Iterations	Statement	Scalar	Vector	Ratio
1000	¯3○R100	.227	.145	1.57
1000	¯3○R200	.461	.251	1.84
1000	¯3○R400	.914	.474	1.93
500	¯3○R800	1.786	.918	1.95
400	¯3○R1600	3.530	2.375	1.49
300	¯3○R3200	6.763	5.287	1.28
200	¯3○R6400	13.210	10.965	1.20
100	¯3○R12800	26.330	22.650	1.16
			Average	1.55

Iterations	Statement	Scalar	Vector	Ratio
500	¯4○B100M	.074	.064	1.16
500	¯4○B200M	.106	.072	1.47
500	¯4○B400M	.188	.114	1.65
200	¯4○B800M	.445	.265	1.68
200	¯4○B1600M	.790	.460	1.72
50	¯4○B3200M	1.500	.760	1.97
50	¯4○B6400M	2.900	1.500	1.93
50	¯4○B12800M	5.640	2.760	2.04
			Average	1.70

Iterations	Statement	Scalar	Vector	Ratio
500	¯4○I100	.394	.124	3.18
500	¯4○I200	.758	.198	3.83
500	¯4○I400	1.502	.382	3.93
200	¯4○I800	2.965	.730	4.06
200	¯4○I1600	6.020	1.525	3.95
50	¯4○I3200	11.920	2.940	4.05
50	¯4○I6400	23.740	5.780	4.11
50	¯4○I12800	47.340	11.380	4.16
			Average	3.91

Iterations	Statement	Scalar	Vector	Ratio
1000	¯4oR100	.354	.119	2.97
1000	¯4oR200	.686	.192	3.57
1000	¯4oR400	1.356	.368	3.68
500	¯4oR800	2.702	.710	3.81
400	¯4oR1600	5.510	1.500	3.67
300	¯4oR3200	10.867	2.907	3.74
200	¯4oR6400	21.685	5.680	3.82
100	¯4oR12800	43.270	11.220	3.86
			Average	3.64

Iterations	Statement	Scalar	Vector	Ratio
500	¯5oB100M	.064	.042	1.52
500	¯5oB200M	.090	.048	1.88
500	¯5oB400M	.156	.066	2.36
200	¯5oB800M	.275	.095	2.89
200	¯5oB1600M	.615	.215	2.86
50	¯5oB3200M	1.160	.360	3.22
50	¯5oB6400M	2.240	.680	3.29
50	¯5oB12800M	4.340	1.320	3.29
			Average	2.67

Iterations	Statement	Scalar	Vector	Ratio
500	¯6oB100M	.090	.080	1.13
500	¯6oB200M	.126	.088	1.43
500	¯6oB400M	.206	.132	1.56
200	¯6oB800M	.440	.290	1.52
200	¯6oB1600M	.810	.450	1.80
50	¯6oB3200M	1.520	.840	1.81
50	¯6oB6400M	2.920	1.480	1.97
50	¯6oB12800M	5.680	2.920	1.95
			Average	1.65

Iterations	Statement	Scalar	Vector	Ratio
500	¯6oI100	.602	.188	3.20
500	¯6oI200	1.178	.322	3.66
500	¯6oI400	2.328	.622	3.74
200	¯6oI800	4.635	1.200	3.86
200	¯6oI1600	9.370	2.505	3.74
50	¯6oI3200	18.560	4.800	3.87
50	¯6oI6400	37.140	9.440	3.93
50	¯6oI12800	73.980	18.720	3.95
			Average	3.74

Iterations	Statement	Scalar	Vector	Ratio
1000	¯6oR100	.565	.186	3.04
1000	¯6oR200	1.108	.312	3.55
1000	¯6oR400	2.199	.612	3.59
500	¯6oR800	4.378	1.196	3.66
400	¯6oR1600	8.860	2.435	3.64
300	¯6oR3200	17.553	4.753	3.69
200	¯6oR6400	35.080	9.335	3.76
100	¯6oR12800	69.880	18.590	3.76
			Average	3.59

Iterations	Statement	Scalar	Vector	Ratio
500	¯7○B100Z	.056	.030	1.87
500	¯7○B200Z	.090	.034	2.65
500	¯7○B400Z	.168	.046	3.65
200	¯7○B800Z	.310	.070	4.43
200	¯7○B1600Z	.695	.185	3.76
50	¯7○B3200Z	1.340	.340	3.94
50	¯7○B6400Z	2.560	.540	4.74
50	¯7○B12800Z	5.080	1.060	4.79
			Average	3.73

Iterations	Statement	Scalar	Vector	Ratio
500	¯8○B100M	.066	.052	1.27
500	¯8○B200M	.098	.062	1.58
500	¯8○B400M	.184	.104	1.77
200	¯8○B800M	.440	.270	1.63
200	¯8○B1600M	.785	.420	1.87
50	¯8○B3200M	1.500	.820	1.83
50	¯8○B6400M	2.920	1.460	2.00
50	¯8○B12800M	5.660	2.820	2.01
			Average	1.74

Iterations	Statement	Scalar	Vector	Ratio
1000	¯10○C100	.050	.038	1.32
1000	¯10○C200	.079	.049	1.61
1000	¯10○C400	.162	.095	1.71
500	¯10○C800	.396	.254	1.56
400	¯10○C1600	.768	.483	1.59
300	¯10○C3200	1.707	1.120	1.52
200	¯10○C6400	3.090	1.955	1.58
100	¯10○C12800	6.040	3.800	1.59
			Average	1.56

Iterations	Statement	Scalar	Vector	Ratio
500	¯11○B100M	.054	.042	1.29
500	¯11○B200M	.086	.050	1.72
500	¯11○B400M	.170	.094	1.81
200	¯11○B800M	.425	.230	1.85
200	¯11○B1600M	.775	.395	1.96
50	¯11○B3200M	1.500	.800	1.88
50	¯11○B6400M	2.840	1.540	1.84
50	¯11○B12800M	5.640	2.800	2.01
			Average	1.79

Iterations	Statement	Scalar	Vector	Ratio
500	¯11○I100	.094	.046	2.04
500	¯11○I200	.160	.060	2.67
500	¯11○I400	.318	.116	2.74
200	¯11○I800	.720	.290	2.48
200	¯11○I1600	1.400	.520	2.69
50	¯11○I3200	2.700	.980	2.76
50	¯11○I6400	5.340	1.940	2.75
50	¯11○I12800	10.520	3.780	2.78
			Average	2.61

Iterations	Statement	Scalar	Vector	Ratio
500	¯11○R100	.068	.040	1.70
500	¯11○R200	.112	.054	2.07
500	¯11○R400	.222	.106	2.09
200	¯11○R800	.510	.275	1.85
200	¯11○R1600	.995	.480	2.07
50	¯11○R3200	1.920	.900	2.13
50	¯11○R6400	3.840	1.780	2.16
50	¯11○R12800	7.480	3.480	2.15
			Average	2.03

Iterations	Statement	Scalar	Vector	Ratio
2000	¯11○C100	.062	.044	1.41
2000	¯11○C200	.099	.058	1.71
500	¯11○C400	.200	.110	1.82
300	¯11○C800	.467	.287	1.63
200	¯11○C1600	.935	.560	1.67
100	¯11○C3200	1.820	1.070	1.70
100	¯11○C6400	3.600	2.020	1.78
50	¯11○C12800	7.240	3.940	1.84
			Average	1.69

Iterations	Statement	Scalar	Vector	Ratio
500	¯12○B100M	.064	.050	1.28
500	¯12○B200M	.092	.058	1.59
500	¯12○B400M	.180	.100	1.80
200	¯12○B800M	.435	.255	1.71
200	¯12○B1600M	.765	.435	1.76
50	¯12○B3200M	1.460	.740	1.97
50	¯12○B6400M	2.860	1.460	1.96
50	¯12○B12800M	5.640	2.800	2.01
			Average	1.76

Iterations	Statement	Scalar	Vector	Ratio
500	¯12○I100	.774	.340	2.28
500	¯12○I200	1.520	.578	2.63
500	¯12○I400	3.028	1.196	2.53
200	¯12○I800	6.135	2.385	2.57
200	¯12○I1600	12.100	4.610	2.62
50	¯12○I3200	24.100	9.040	2.67
50	¯12○I6400	48.140	18.040	2.67
50	¯12○I12800	96.300	35.540	2.71
			Average	2.58

Iterations	Statement	Scalar	Vector	Ratio
500	¯12○R100	.590	.336	1.76
500	¯12○R200	1.208	.584	2.07
500	¯12○R400	2.644	1.156	2.29
200	¯12○R800	5.595	2.390	2.34
200	¯12○R1600	11.355	4.670	2.43
50	¯12○R3200	22.860	9.180	2.49
50	¯12○R6400	45.960	18.240	2.52
50	¯12○R12800	91.800	36.540	2.51
			Average	2.30

Iterations	Statement	Scalar	Vector	Ratio
2000	¯12○C100	.826	.313	2.64
2000	¯12○C200	1.619	.554	2.92
500	¯12○C400	3.228	1.120	2.88
300	¯12○C800	6.570	2.323	2.83
200	¯12○C1600	13.005	4.510	2.88
100	¯12○C3200	25.870	8.930	2.90
100	¯12○C6400	51.590	17.700	2.91
50	¯12○C12800	103.420	35.240	2.93
			Average	2.86

Non-Scalar Primitives

Iterations	Statement	Scalar	Vector	Ratio
3000	+/B100M	.031	.027	1.15
3000	+/B200M	.035	.027	1.29
2000	+/B400M	.046	.029	1.59
500	+/B800M	.068	.034	2.00
200	+/B1600M	.105	.040	2.63
200	+/B3200M	.195	.050	3.90
100	+/B6400M	.350	.080	4.38
100	+/B12800M	.680	.130	5.23
			Average	2.77

Iterations	Statement	Scalar	Vector	Ratio
3000	+/R100	.034	.034	1.00
3000	+/R200	.041	.037	1.11
2000	+/R400	.057	.043	1.31
500	+/R800	.086	.058	1.48
200	+/R1600	.150	.085	1.76
200	+/R3200	.265	.135	1.96
100	+/R6400	.520	.260	2.00
100	+/R12800	1.040	.490	2.12
			Average	1.59

Iterations	Statement	Scalar	Vector	Ratio
50	+/[1]I100_100	.920	.420	2.19
50	+/[1]I200_200	3.960	1.620	2.44
50	+/[1]I400_400	13.720	6.380	2.15
			Average	2.26

Iterations	Statement	Scalar	Vector	Ratio
50	+/[2]I100_100	.940	.420	2.24
50	+/[2]I200_200	3.620	3.620	1.00
50	+/[2]I400_400	13.780	13.760	1.00
			Average	1.41

Iterations	Statement	Scalar	Vector	Ratio
50	+/[1]R100_100	.960	.440	2.18
50	+/[1]R200_200	3.940	1.640	2.40
50	+/[1]R400_400	25.040	6.320	3.96
			Average	2.85

Iterations	Statement	Scalar	Vector	Ratio
50	+/[2]R100_100	.980	.660	1.48
50	+/[2]R200_200	3.980	2.080	1.91
50	+/[2]R400_400	15.060	7.100	2.12
			Average	1.84

Iterations	Statement	Scalar	Vector	Ratio
50	+/[2]I100_2	.300	.100	3.00
50	+/[2]I200_2	.540	.100	5.40
50	+/[2]I400_2	1.020	.160	6.38
50	+/[2]I800_2	1.980	.280	7.07
50	+/[2]I1600_2	3.980	.540	7.37
50	+/[2]I3200_2	7.920	.980	8.08
50	+/[2]I6400_2	15.780	1.920	8.22
50	+/[2]I12800_2	31.480	3.760	8.37
			Average	6.74

Iterations	Statement	Scalar	Vector	Ratio
50	+/[2]R100_2	.200	.060	3.33
50	+/[2]R200_2	3.720	.820	4.54
50	+/[2]R400_2	.700	.120	5.83
50	+/[2]R800_2	1.380	.220	6.27
50	+/[2]R1600_2	2.780	.460	6.04
50	+/[2]R3200_2	5.540	.980	5.65
50	+/[2]R6400_2	11.220	1.800	6.23
50	+/[2]R12800_2	22.120	3.500	6.32
			Average	5.53

Iterations	Statement	Scalar	Vector	Ratio
1000	+/[1]I10_2_2	.040	.041	.98
1000	+/[1]I100_2_2	.067	.067	1.00
1000	+/[1]I500_2_2	.187	.188	.99
1000	+/[1]I10_5_5	.085	.054	1.57
1000	+/[1]I100_5_5	.256	.196	1.31
1000	+/[1]I500_5_5	1.008	.823	1.22
500	+/[1]I10_10_10	.238	.078	3.05
500	+/[1]I100_10_10	.914	.426	2.15
500	+/[1]I500_10_10	5.176	2.300	2.25
100	+/[1]I10_20_20	.850	.180	4.72
100	+/[1]I100_20_20	3.800	1.590	2.39
100	+/[1]I500_20_20	17.650	8.110	2.18
			Average	1.98

Iterations	Statement	Scalar	Vector	Ratio
500	+/[1]R10_2_2	.042	.044	.95
500	+/[1]R100_2_2	.072	.064	1.13
500	+/[1]R500_2_2	.190	.116	1.64
500	+/[1]R10_5_5	.084	.054	1.56
500	+/[1]R100_5_5	.256	.188	1.36
500	+/[1]R500_5_5	1.054	.814	1.29
400	+/[1]R10_10_10	.240	.080	3.00
400	+/[1]R100_10_10	.950	.435	2.18
400	+/[1]R500_10_10	7.123	2.358	3.02
100	+/[1]R10_20_20	.860	.180	4.78
100	+/[1]R100_20_20	4.200	1.690	2.49
100	+/[1]R500_20_20	40.440	8.400	4.81
			Average	2.35

Iterations	Statement	Scalar	Vector	Ratio
1000	+/[1]C10_2_2	.044	.044	1.00
1000	+/[1]C100_2_2	.087	.087	1.00
1000	+/[1]C500_2_2	.274	.273	1.00
500	+/[1]C10_5_5	.096	.074	1.30
500	+/[1]C100_5_5	.348	.368	.95
500	+/[1]C500_5_5	3.052	2.080	1.47
400	+/[1]C10_10_10	.280	.128	2.20
400	+/[1]C100_10_10	1.523	1.108	1.37
400	+/[1]C500_10_10	7.673	5.340	1.44
100	+/[1]C10_20_20	1.040	.370	2.81
100	+/[1]C100_20_20	6.430	4.040	1.59
100	+/[1]C500_20_20	60.080	20.140	2.98
			Average	1.59

Iterations	Statement	Scalar	Vector	Ratio
1000	+/[2]I2_10_2	.041	.042	.98
1000	+/[2]I2_100_2	.067	.068	.99
1000	+/[2]I2_500_2	.187	.188	.99
1000	+/[2]I5_10_5	.086	.087	.99
1000	+/[2]I5_100_5	.255	.256	1.00
1000	+/[2]I5_500_5	1.015	1.016	1.00
500	+/[2]I10_10_10	.240	.242	.99
500	+/[2]I10_100_10	.932	.930	1.00
500	+/[2]I10_500_10	4.382	4.400	1.00
100	+/[2]I20_10_20	.850	.340	2.50
100	+/[2]I20_100_20	3.900	3.300	1.18
100	+/[2]I20_500_20	17.280	16.330	1.06
			Average	1.14

Iterations	Statement	Scalar	Vector	Ratio
500	+/[2]R2_10_2	.044	.044	1.00
500	+/[2]R2_100_2	.070	.062	1.13
500	+/[2]R2_500_2	.194	.118	1.64
500	+/[2]R5_10_5	.086	.086	1.00
500	+/[2]R5_100_5	.258	.198	1.30
500	+/[2]R5_500_5	1.066	.606	1.76
400	+/[2]R10_10_10	.240	.240	1.00
400	+/[2]R10_100_10	.970	.728	1.33
400	+/[2]R10_500_10	4.810	3.218	1.49
100	+/[2]R20_10_20	.860	.340	2.53
100	+/[2]R20_100_20	4.210	3.480	1.21
100	+/[2]R20_500_20	18.980	16.480	1.15
			Average	1.38

Iterations	Statement	Scalar	Vector	Ratio
1000	+/[2]C2_10_2	.049	.050	.98
1000	+/[2]C2_100_2	.092	.092	1.00
1000	+/[2]C2_500_2	.305	.306	1.00
500	+/[2]C5_10_5	.110	.114	.96
500	+/[2]C5_100_5	.388	.390	.99
500	+/[2]C5_500_5	2.148	2.202	.98
400	+/[2]C10_10_10	.303	.315	.96
400	+/[2]C10_100_10	1.758	1.778	.99
400	+/[2]C10_500_10	8.475	8.610	.98
100	+/[2]C20_10_20	1.160	.820	1.41
100	+/[2]C20_100_20	6.810	7.780	.88
100	+/[2]C20_500_20	34.440	38.400	.90
			Average	1.00

Iterations	Statement	Scalar	Vector	Ratio
1000	+/[3]I2_2_10	.041	.041	1.00
1000	+/[3]I2_2_100	.068	.068	1.00
1000	+/[3]I2_2_500	.188	.188	1.00
1000	+/[3]I5_5_10	.090	.058	1.55
1000	+/[3]I5_5_100	.259	.185	1.40
1000	+/[3]I5_5_500	1.023	1.025	1.00
500	+/[3]I10_10_10	.260	.084	3.10
500	+/[3]I10_10_100	.940	.418	2.25
500	+/[3]I10_10_500	4.308	4.334	.99
100	+/[3]I20_20_10	.940	.200	4.70
100	+/[3]I20_20_100	3.940	1.810	2.18
100	+/[3]I20_20_500	17.030	17.100	1.00
			Average	1.76

Iterations	Statement	Scalar	Vector	Ratio
500	+/[3]R2_2_10	.042	.044	.95
500	+/[3]R2_2_100	.072	.062	1.16
500	+/[3]R2_2_500	.190	.116	1.64
500	+/[3]R5_5_10	.090	.060	1.50
500	+/[3]R5_5_100	.260	.184	1.41
500	+/[3]R5_5_500	1.070	.552	1.94
400	+/[3]R10_10_10	.263	.088	3.00
400	+/[3]R10_10_100	.958	.640	1.50
400	+/[3]R10_10_500	4.710	2.245	2.10
100	+/[3]R20_20_10	.960	.210	4.57
100	+/[3]R20_20_100	4.260	2.680	1.59
100	+/[3]R20_20_500	18.640	8.490	2.20
			Average	1.96

Iterations	Statement	Scalar	Vector	Ratio
1000	+/[3]C2_2_10	.044	.045	.98
1000	+/[3]C2_2_100	.085	.084	1.01
1000	+/[3]C2_2_500	.274	.274	1.00
500	+/[3]C5_5_10	.102	.072	1.42
500	+/[3]C5_5_100	.350	.354	.99
500	+/[3]C5_5_500	1.906	1.948	.98
400	+/[3]C10_10_10	.310	.128	2.43
400	+/[3]C10_10_100	1.643	1.643	1.00
400	+/[3]C10_10_500	7.148	7.193	.99
100	+/[3]C20_20_10	1.140	.350	3.26
100	+/[3]C20_20_100	6.060	6.160	.98
100	+/[3]C20_20_500	29.550	30.110	.98
			Average	1.34

Iterations	Statement	Scalar	Vector	Ratio
1	I100_100+.×I100_100	246.000	187.000	1.32
1	I200_200+.×I200_200	2030.000	1398.000	1.45
1	I400_400+.×I400_400	16030.000	11338.000	1.41
			Average	1.39

Iterations	Statement	Scalar	Vector	Ratio
1	R100_100+.×R100_100	191.000	23.000	8.30
1	R200_200+.×R200_200	1630.000	203.000	8.03
1	R400_400+.×R400_400	17874.000	1664.000	10.74
			Average	9.03

Iterations	Statement	Scalar	Vector	Ratio
1	C100_100+.×C100_100	854.000	328.000	2.60
1	C200_200+.×C200_200	6860.000	2418.000	2.84
1	C400_400+.×C400_400	63453.000	19550.000	3.25
			Average	2.90

Iterations	Statement	Scalar	Vector	Ratio
50	I20_20+.×B20_20	6.460	.740	8.73
50	I50_50+.×B50_50	97.540	5.180	18.83
1	I100_100+.×B100_100	773.000	27.000	28.63
1	I200_200+.×B200_200	6149.000	166.000	37.04
1	I400_400+.×B400_400	49109.000	1318.000	37.26
			Average	26.10

Iterations	Statement	Scalar	Vector	Ratio
10	B20_20+.×I20_20	6.300	.500	12.60
10	B50_50+.×I50_50	94.800	3.100	30.58
1	B100_100+.×I100_100	753.000	16.000	47.06
1	B200_200+.×I200_200	6237.000	107.000	58.29
1	B400_400+.×I400_400	49619.000	941.000	52.73
			Average	40.25

Iterations	Statement	Scalar	Vector	Ratio
50	R20_20+.×B20_20	6.020	.740	8.14
50	R50_50+.×B50_50	90.780	5.240	17.32
1	R100_100+.×B100_100	718.000	27.000	26.59
1	R200_200+.×B200_200	5703.000	167.000	34.15
1	R400_400+.×B400_400	45532.000	1321.000	34.47
			Average	24.13

Iterations	Statement	Scalar	Vector	Ratio
10	B20_20+.×R20_20	6.000	.400	15.00
10	B50_50+.×R50_50	90.900	3.100	29.32
1	B100_100+.×R100_100	731.000	16.000	45.69
1	B200_200+.×R200_200	5946.000	129.000	46.09
1	B400_400+.×R400_400	56071.000	999.000	56.13
			Average	38.45

Iterations	Statement	Scalar	Vector	Ratio
2	I100•.×I100D	2.500	1.500	1.67
2	I200•.×I200D	9.000	6.000	1.50
1	I400•.×I400D	34.000	21.000	1.62
			Average	1.60

Iterations	Statement	Scalar	Vector	Ratio
2	R100•.×R100D	2.000	1.000	2.00
2	R200•.×R200D	7.500	3.500	2.14
1	R400•.×R400D	27.000	10.000	2.70
			Average	2.28

Iterations	Statement	Scalar	Vector	Ratio
1	C100•.×C100D	6.000	3.000	2.00
1	C200•.×C200D	23.000	11.000	2.09
			Average	2.05

Iterations	Statement	Scalar	Vector	Ratio
2	I100•.+I100D	1.500	1.000	1.50
2	I200•.+I200D	5.500	2.500	2.20
1	I400•.+I400D	18.000	9.000	2.00
			Average	1.90

Iterations	Statement	Scalar	Vector	Ratio
2	R100•.+R100D	2.000	1.000	2.00
2	R200•.+R200D	6.500	3.500	1.86
1	R400•.+R400D	24.000	10.000	2.40
			Average	2.09

Iterations	Statement	Scalar	Vector	Ratio
1	C100•.+C100D	3.000	2.000	1.50
1	C200•.+C200D	11.000	6.000	1.83
			Average	1.67

Iterations	Statement	Scalar	Vector	Ratio
150	MAPE100\I100	.147	.040	3.67
150	MAPE200\I200	.267	.060	4.44
150	MAPE400\I400	.520	.093	5.57
150	MAPE800\I800	1.013	.153	6.61
150	MAPE1600\I1600	2.087	.360	5.80
75	MAPE3200\I3200	4.120	.653	6.31
75	MAPE6400\I6400	8.200	1.360	6.03
75	MAPE12800\I12800	16.440	2.520	6.52
			Average	5.62

Iterations	Statement	Scalar	Vector	Ratio
150	MAPE100\R100	.160	.040	4.00
150	MAPE200\R200	.293	.060	4.89
150	MAPE400\R400	.580	.100	5.80
150	MAPE800\R800	1.213	.247	4.92
150	MAPE1600\R1600	2.373	.433	5.48
75	MAPE3200\R3200	4.693	.827	5.68
50	MAPE6400\R6400	9.480	1.620	5.85
50	MAPE12800\R12800	18.820	3.200	5.88
			Average	5.31

Iterations	Statement	Scalar	Vector	Ratio
100	MAPE100\C100	.230	.060	3.83
100	MAPE200\C200	.440	.090	4.89
100	MAPE400\C400	.950	.260	3.65
50	MAPE800\C800	1.860	.460	4.04
50	MAPE1600\C1600	3.680	.880	4.18
25	MAPE3200\C3200	7.360	1.720	4.28
20	MAPE6400\C6400	14.550	3.250	4.48
20	MAPE12800\C12800	29.000	6.400	4.53
			Average	4.24

Iterations	Statement	Scalar	Vector	Ratio
100	B100M,I100	.050	.040	1.25
100	B200M,I200	.080	.050	1.60
100	B400M,I400	.140	.070	2.00
100	B800M,I800	.270	.100	2.70
50	B1600M,I1600	.600	.240	2.50
50	B3200M,I3200	1.260	.420	3.00
25	B6400M,I6400	2.200	.760	2.89
15	B12800M,I12800	4.467	1.600	2.79
			Average	2.34

Iterations	Statement	Scalar	Vector	Ratio
100	B100M,R100	.060	.050	1.20
50	B200M,R200	.080	.060	1.33
50	B400M,R400	.160	.100	1.60
25	B800M,R800	.400	.200	2.00
25	B1600M,R1600	.680	.360	1.89
10	B3200M,R3200	1.300	.600	2.17
15	B6400M,R6400	2.667	1.400	1.90
5	B12800M,R12800	5.200	2.600	2.00
			Average	1.76

Iterations	Statement	Scalar	Vector	Ratio
50	B100M,C100	.080	.040	2.00
50	B200M,C200	.120	.060	2.00
25	B400M,C400	.320	.200	1.60
25	B800M,C800	.520	.360	1.44
15	B1600M,C1600	1.067	.667	1.60
15	B3200M,C3200	2.067	1.400	1.48
5	B6400M,C6400	4.000	2.600	1.54
5	B12800M,C12800	7.800	4.800	1.63
			Average	1.66

Iterations	Statement	Scalar	Vector	Ratio
100	I100,R100	.070	.040	1.75
100	I200,R200	.100	.050	2.00
50	I400,R400	.200	.080	2.50
50	I800,R800	.440	.220	2.00
25	I1600,R1600	.800	.400	2.00
25	I3200,R3200	1.520	.720	2.11
15	I6400,R6400	3.133	1.467	2.14
5	I12800,R12800	5.800	2.600	2.23
			Average	2.09

Iterations	Statement	Scalar	Vector	Ratio
100	I100,C100	.080	.050	1.60
50	I200,C200	.120	.080	1.50
50	I400,C400	.340	.220	1.55
25	I800,C800	.600	.360	1.67
20	I1600,C1600	1.150	.700	1.64
15	I3200,C3200	2.333	1.400	1.67
10	I6400,C6400	4.500	2.700	1.67
5	I12800,C12800	8.800	5.000	1.76
			Average	1.63

Iterations	Statement	Scalar	Vector	Ratio
50	R100,C100	.060	.040	1.50
25	R200,C200	.080	.080	1.00
25	R400,C400	.240	.200	1.20
20	R800,C800	.400	.350	1.14
15	R1600,C1600	.867	.667	1.30
20	R3200,C3200	1.700	1.350	1.26
10	R6400,C6400	3.200	2.600	1.23
5	R12800,C12800	6.200	5.000	1.24
			Average	1.23

Iterations	Statement	Scalar	Vector	Ratio
100	J100,I100	.040	.040	1.00
100	J200,I200	.050	.040	1.25
100	J400,I400	.080	.070	1.14
100	J800,I800	.140	.090	1.56
50	J1600,I1600	.320	.240	1.33
50	J3200,I3200	.580	.400	1.45
25	J6400,I6400	1.120	.760	1.47
15	J12800,I12800	2.333	1.533	1.52
			Average	1.34

Iterations	Statement	Scalar	Vector	Ratio
100	J100,R100	.050	.050	1.00
100	J200,R200	.050	.050	1.00
100	J400,R400	.110	.100	1.10
100	J800,R800	.260	.230	1.13
50	J1600,R1600	.460	.400	1.15
50	J3200,R3200	.860	.760	1.13
25	J6400,R6400	1.800	1.520	1.18
15	J12800,R12800	3.400	2.933	1.16
			Average	1.11

Iterations	Statement	Scalar	Vector	Ratio
100	J100,C100	.060	.050	1.20
100	J200,C200	.080	.070	1.14
100	J400,C400	.260	.220	1.18
100	J800,C800	.430	.380	1.13
50	J1600,C1600	.840	.720	1.17
50	J3200,C3200	1.680	1.580	1.06
25	J6400,C6400	3.240	2.760	1.17
15	J12800,C12800	6.267	5.400	1.16
			Average	1.15

Iterations	Statement	Scalar	Vector	Ratio
30	⍉I20_20	.133	.100	1.33
30	⍉I50_50	.467	.233	2.00
25	⍉I100_100	1.840	.920	2.00
15	⍉I200_200	7.867	4.000	1.97
5	⍉I400_400	27.200	11.600	2.34
			Average	1.93

Iterations	Statement	Scalar	Vector	Ratio
40	⍉R20_20	.100	.100	1.00
40	⍉R50_50	.425	.250	1.70
20	⍉R100_100	2.200	1.300	1.69
10	⍉R200_200	8.100	4.400	1.84
3	⍉R400_400	42.000	29.333	1.43
			Average	1.53

Iterations	Statement	Scalar	Vector	Ratio
20	⍉C20_20	.150	.100	1.50
20	⍉C50_50	1.150	.500	2.30
10	⍉C100_100	3.500	2.700	1.30
5	⍉C200_200	13.400	9.200	1.46
2	⍉C400_400	80.000	75.500	1.06
			Average	1.52

Iterations	Statement	Scalar	Vector	Ratio
50	⌽[2]I20_20	.140	.120	1.17
50	⌽[2]I50_50	.460	.280	1.64
40	⌽[2]I100_100	1.875	.950	1.97
15	⌽[2]I200_200	7.067	3.400	2.08
5	⌽[2]I400_400	26.600	12.200	2.18
			Average	1.81

Iterations	Statement	Scalar	Vector	Ratio
40	⌽[2]R20_20	.125	.125	1.00
30	⌽[2]R50_50	.633	.300	2.11
20	⌽[2]R100_100	2.300	1.500	1.53
10	⌽[2]R200_200	8.200	5.000	1.64
5	⌽[2]R400_400	31.400	18.400	1.71
			Average	1.60

Iterations	Statement	Scalar	Vector	Ratio
20	Φ[2]C20_20	.150	.150	1.00
10	Φ[2]C50_50	1.000	.700	1.43
5	Φ[2]C100_100	3.600	2.800	1.29
2	Φ[2]C200_200	12.500	9.000	1.39
1	Φ[2]C400_400	46.000	30.000	1.53
			Average	1.33

Iterations	Statement	Scalar	Vector	Ratio
500	2-/I100	.154	.036	4.28
500	2-/I200	.280	.040	7.00
250	2-/I400	.520	.056	9.29
100	2-/I800	1.010	.080	12.63
100	2-/I1600	1.990	.140	14.21
50	2-/I3200	4.060	.320	12.69
25	2-/I6400	8.080	.600	13.47
25	2-/I12800	16.120	1.160	13.90
			Average	10.93

Iterations	Statement	Scalar	Vector	Ratio
500	2-/R100	.156	.036	4.33
500	2-/R200	.276	.042	6.57
250	2-/R400	.532	.060	8.87
100	2-/R800	1.040	.090	11.56
100	2-/R1600	2.170	.230	9.43
50	2-/R3200	4.220	.440	9.59
25	2-/R6400	8.400	.840	10.00
25	2-/R12800	16.680	1.520	10.97
			Average	8.92

Iterations	Statement	Scalar	Vector	Ratio
25	2-/[1]I100_100	10.240	.760	13.47
10	2-/[1]I200_200	40.600	3.500	11.60
10	2-/[1]I400_400	161.800	13.400	12.07
			Average	12.38

Iterations	Statement	Scalar	Vector	Ratio
25	2-/[2]I100_100	12.160	.800	15.20
10	2-/[2]I200_200	49.700	3.500	14.20
10	2-/[2]I400_400	198.600	13.500	14.71
			Average	14.70

Iterations	Statement	Scalar	Vector	Ratio
25	2-/[1]R100_100	10.360	.960	10.79
10	2-/[1]R200_200	42.500	4.500	9.44
10	2-/[1]R400_400	170.500	17.900	9.53
			Average	9.92

Iterations	Statement	Scalar	Vector	Ratio
25	2-/[2]R100_100	12.640	1.000	12.64
10	2-/[2]R200_200	51.900	4.400	11.80
10	2-/[2]R400_400	206.900	17.300	11.96
			Average	12.13

Iterations	Statement	Scalar	Vector	Ratio
500	2+/I100	.156	.034	4.59
500	2+/I200	.276	.042	6.57
250	2+/I400	.520	.056	9.29
100	2+/I800	1.020	.080	12.75
100	2+/I1600	2.000	.130	15.38
50	2+/I3200	4.100	.320	12.81
25	2+/I6400	8.080	.600	13.47
25	2+/I12800	16.080	1.160	13.86
			Average	11.09

Iterations	Statement	Scalar	Vector	Ratio
500	2+/R100	.160	.038	4.21
500	2+/R200	.282	.040	7.05
250	2+/R400	.540	.060	9.00
100	2+/R800	1.040	.090	11.56
100	2+/R1600	2.160	.240	9.00
50	2+/R3200	4.260	.400	10.65
25	2+/R6400	8.400	.840	10.00
25	2+/R12800	16.760	1.720	9.74
			Average	8.90

Iterations	Statement	Scalar	Vector	Ratio
25	2+/[1]I100_100	10.280	.760	13.53
10	2+/[1]I200_200	40.600	3.500	11.60
10	2+/[1]I400_400	161.900	13.100	12.36
			Average	12.50

Iterations	Statement	Scalar	Vector	Ratio
25	2+/[2]I100_100	12.280	.760	16.16
10	2+/[2]I200_200	49.900	3.400	14.68
10	2+/[2]I400_400	198.700	13.600	14.61
			Average	15.15

Iterations	Statement	Scalar	Vector	Ratio
25	2+/[1]R100_100	10.320	1.000	10.32
10	2+/[1]R200_200	42.400	4.500	9.42
10	2+/[1]R400_400	169.700	17.800	9.53
			Average	9.76

Iterations	Statement	Scalar	Vector	Ratio
25	2+/[2]R100_100	12.640	.920	13.74
10	2+/[2]R200_200	51.500	4.800	10.73
10	2+/[2]R400_400	206.200	19.300	10.68
			Average	11.72

Iterations	Statement	Scalar	Vector	Ratio
2000	I12800[IN20]	.033	.034	.97
2000	I12800[IN50]	.039	.036	1.08
1000	I12800[IN100]	.052	.041	1.27
1000	I12800[IN200]	.078	.052	1.50
500	I12800[IN400]	.128	.074	1.73
500	I12800[IN800]	.232	.122	1.90
250	I12800[IN1600]	.436	.212	2.06
250	I12800[IN3200]	.988	.508	1.94
100	I12800[IN6400]	1.900	.990	1.92
100	I12800[IN12800]	3.900	1.950	2.00
			Average	1.64

Iterations	Statement	Scalar	Vector	Ratio
2000	R12800[IN20]	.032	.034	.94
2000	R12800[IN50]	.039	.037	1.07
1000	R12800[IN100]	.052	.042	1.24
1000	R12800[IN200]	.076	.055	1.38
500	R12800[IN400]	.128	.082	1.56
500	R12800[IN800]	.230	.136	1.69
250	R12800[IN1600]	.548	.352	1.56
250	R12800[IN3200]	1.052	.652	1.61
100	R12800[IN6400]	2.110	1.270	1.66
100	R12800[IN12800]	4.720	2.670	1.77
			Average	1.45

Iterations	Statement	Scalar	Vector	Ratio
1000	I100_100[;5]	.147	.048	3.06
500	I200_200[;5]	.276	.072	3.83
100	I400_400[;5]	.470	.080	5.88
			Average	4.26

Iterations	Statement	Scalar	Vector	Ratio
1000	R100_100[;5]	.148	.051	2.90
500	R200_200[;5]	.260	.060	4.33
100	R400_400[;5]	.560	.130	4.31
			Average	3.85

Iterations	Statement	Scalar	Vector	Ratio
2000	I10_10[;I5N]	.064	.056	1.15
1000	I20_20[;I10N]	.114	.070	1.63
1000	I50_50[;I25N]	.424	.157	2.70
1000	I100_100[;I50N]	1.688	.713	2.37
500	I200_200[;I100N]	6.118	2.670	2.29
100	I400_400[;I200N]	23.900	9.920	2.41
			Average	2.09

Iterations	Statement	Scalar	Vector	Ratio
2000	R10_10[;I5N]	.063	.055	1.15
1000	R20_20[;I10N]	.109	.070	1.56
1000	R50_50[;I25N]	.500	.160	3.13
1000	R100_100[;I50N]	1.730	.824	2.10
500	R200_200[;I100N]	6.432	3.368	1.91
100	R400_400[;I200N]	25.570	12.520	2.04
			Average	1.98

Iterations	Statement	Scalar	Vector	Ratio
2000	I12800[IN20]←1234	.036	.037	.99
2000	I12800[IN50]←1234	.046	.040	1.14
1000	I12800[IN100]←1234	.059	.043	1.37
1000	I12800[IN200]←1234	.092	.053	1.74
500	I12800[IN400]←1234	.148	.068	2.18
500	I12800[IN800]←1234	.268	.082	3.27
250	I12800[IN1600]1234	.460	.240	1.92
250	I12800[IN3200]←1234	1.028	.324	3.17
100	I12800[IN6400]←1234	2.040	.880	2.32
100	I12800[IN12800]←1234	4.030	1.880	2.14
			Average	2.02

Iterations	Statement	Scalar	Vector	Ratio
2000	R12800[IN20]←.1234	.036	.037	.97
2000	R12800[IN50]←.1234	.045	.040	1.13
1000	R12800[IN100]←.1234	.059	.044	1.34
1000	R12800[IN200]←.1234	.089	.052	1.71
500	R12800[IN400]←.1234	.148	.062	2.39
500	R12800[IN800]←.1234	.266	.080	3.33
250	R12800[IN1600]←.1234	.512	.188	2.72
250	R12800[IN3200]←.1234	1.016	.444	2.29
100	R12800[IN6400]←.1234	2.120	.930	2.28
100	R12800[IN12800]←.1234	4.640	2.130	2.18
			Average	2.03

Idioms

Iterations	Statement	Scalar	Vector	Ratio
15	+/[1]|I100_100	3.600	1.067	3.38
15	+/[1]|I200_200	16.133	5.600	2.88
15	+/[1]|I400_400	62.267	21.267	2.93
			Average	3.06

Iterations	Statement	Scalar	Vector	Ratio
15	+/[2]|I100_100	3.600	1.067	3.38
15	+/[2]|I200_200	15.867	7.600	2.09
15	+/[2]|I400_400	62.133	28.600	2.17
			Average	2.55

Iterations	Statement	Scalar	Vector	Ratio
15	+/[1]|R100_100	3.867	1.200	3.22
10	+/[1]|R200_200	17.900	7.200	2.49
10	+/[1]|R400_400	80.200	27.700	2.90
			Average	2.87

Iterations	Statement	Scalar	Vector	Ratio
15	+/[2]|R100_100	4.000	1.467	2.73
10	+/[2]|R200_200	18.300	7.700	2.38
10	+/[2]|R400_400	71.200	28.500	2.50
			Average	2.53

Iterations	Statement	Scalar	Vector	Ratio
5	+/[1]|C100_100	39.600	14.000	2.83
3	+/[1]|C200_200	160.333	58.000	2.76
3	+/[1]|C400_400	649.333	230.667	2.82
			Average	2.80

Iterations	Statement	Scalar	Vector	Ratio
5	+/[2]|C100_100	39.600	14.200	2.79
3	+/[2]|C200_200	160.000	58.667	2.73
3	+/[2]|C400_400	639.000	231.333	2.76
			Average	2.76

Iterations	Statement	Scalar	Vector	Ratio
25	⌈/[1]|I100_100	4.000	1.160	3.45
15	⌈/[1]|I200_200	18.067	6.133	2.95
10	⌈/[1]|I400_400	69.400	23.800	2.92
			Average	3.10

Iterations	Statement	Scalar	Vector	Ratio
25	⌈/[2]|I100_100	4.080	1.160	3.52
15	⌈/[2]|I200_200	17.800	9.400	1.89
10	⌈/[2]|I400_400	69.500	36.200	1.92
			Average	2.44

Iterations	Statement	Scalar	Vector	Ratio
25	⌈/[1]|R100_100	4.320	1.400	3.09
10	⌈/[1]|R200_200	19.900	7.900	2.52
5	⌈/[1]|R400_400	87.200	30.000	2.91
			Average	2.84

Iterations	Statement	Scalar	Vector	Ratio
25	⌈/[2]|R100_100	4.360	1.920	2.27
10	⌈/[2]|R200_200	20.100	9.700	2.07
5	⌈/[2]|R400_400	78.000	36.600	2.13
			Average	2.16

Iterations	Statement	Scalar	Vector	Ratio
5	⌈/[1]|C100_100	40.200	14.200	2.83
3	⌈/[1]|C200_200	162.000	58.667	2.76
2	⌈/[1]|C400_400	655.500	233.000	2.81
			Average	2.80

Iterations	Statement	Scalar	Vector	Ratio
5	⌈/[2]|C100_100	40.000	14.800	2.70
3	⌈/[2]|C200_200	162.333	61.000	2.66
2	⌈/[2]|C400_400	646.000	241.000	2.68
			Average	2.68

Iterations	Statement	Scalar	Vector	Ratio
25	⌊/[1]|I100_100	3.880	1.200	3.23
15	⌊/[1]|I200_200	17.533	6.133	2.86
10	⌊/[1]|I400_400	67.200	24.500	2.74
			Average	2.94

Iterations	Statement	Scalar	Vector	Ratio
25	⌊/[2]|I100_100	3.960	1.200	3.30
15	⌊/[2]|I200_200	17.400	8.867	1.96
10	⌊/[2]|I400_400	67.600	33.900	1.99
			Average	2.42

Iterations	Statement	Scalar	Vector	Ratio
25	⌊/[1]|R100_100	4.200	1.360	3.09
10	⌊/[1]|R200_200	19.400	7.600	2.55
5	⌊/[1]|R400_400	86.600	29.000	2.99
			Average	2.88

Iterations	Statement	Scalar	Vector	Ratio
25	⌊/[2]|R100_100	4.320	1.960	2.20
10	⌊/[2]|R200_200	19.700	9.800	2.01
5	⌊/[2]|R400_400	76.600	36.600	2.09
			Average	2.10

Iterations	Statement	Scalar	Vector	Ratio
5	⌊/[1]|C100_100	40.200	14.200	2.83
3	⌊/[1]|C200_200	161.667	58.667	2.76
2	⌊/[1]|C400_400	655.000	232.500	2.82
			Average	2.80

Iterations	Statement	Scalar	Vector	Ratio
5	⌊/[2]|C100_100	40.000	15.000	2.67
3	⌊/[2]|C200_200	161.667	60.667	2.66
2	⌊/[2]|C400_400	645.500	240.500	2.68
			Average	2.67

Iterations	Statement	Scalar	Vector	Ratio
1000	R100ι⌈/R100	.028	.027	1.04
1000	R200ι⌈/R200	.038	.037	1.03
1000	R400ι⌈/R400	.059	.055	1.07
1000	R800ι⌈/R800	.101	.093	1.09
1000	R1600ι⌈/R1600	.185	.168	1.10
1000	R3200ι⌈/R3200	.354	.318	1.11
1000	R6400ι⌈/R6400	.691	.616	1.12
1000	R12800ι⌈/R12800	1.389	1.233	1.13
			Average	1.09

Iterations	Statement	Scalar	Vector	Ratio
1000	R100ι⌊/R100	.029	.027	1.07
1000	R200ι⌊/R200	.041	.036	1.14
1000	R400ι⌊/R400	.065	.055	1.18
1000	R800ι⌊/R800	.115	.094	1.22
1000	R1600ι⌊/R1600	.211	.168	1.26
1000	R3200ι⌊/R3200	.402	.318	1.26
1000	R6400ι⌊/R6400	.786	.616	1.28
1000	R12800ι⌊/R12800	1.580	1.238	1.28
			Average	1.21

Iterations	Statement	Scalar	Vector	Ratio
1000	⍎1⍕R20	.056	.051	1.10
1000	⍎1⍕R50	.106	.073	1.45
500	⍎1⍕R100	.190	.108	1.76
500	⍎1⍕R200	.356	.186	1.91
200	⍎1⍕R400	.695	.345	2.01
200	⍎1⍕R800	1.380	.670	2.06
100	⍎1⍕R1600	2.880	1.420	2.03
100	⍎1⍕R3200	5.600	2.730	2.05
100	⍎1⍕R6400	11.020	5.330	2.07
100	⍎1⍕R12800	21.900	10.520	2.08
			Average	1.85

Iterations	Statement	Scalar	Vector	Ratio
1000	⍎2⍕R20	.056	.051	1.10
1000	⍎2⍕R50	.106	.074	1.43
500	⍎2⍕R100	.192	.110	1.75
500	⍎2⍕R200	.370	.188	1.97
200	⍎2⍕R400	.705	.345	2.04
200	⍎2⍕R800	1.385	.665	2.08
100	⍎2⍕R1600	2.910	1.410	2.06
100	⍎2⍕R3200	5.690	2.760	2.06
100	⍎2⍕R6400	11.190	5.340	2.10
100	⍎2⍕R12800	22.310	10.480	2.13
			Average	1.87

Iterations	Statement	Scalar	Vector	Ratio
1000	⍎3⍕R20	.058	.052	1.12
1000	⍎3⍕R50	.113	.075	1.51
500	⍎3⍕R100	.200	.110	1.82
500	⍎3⍕R200	.378	.184	2.05
200	⍎3⍕R400	.725	.350	2.07
200	⍎3⍕R800	1.445	.655	2.21
100	⍎3⍕R1600	3.020	1.410	2.14
100	⍎3⍕R3200	5.940	2.740	2.17
100	⍎3⍕R6400	11.740	5.330	2.20
100	⍎3⍕R12800	23.210	10.490	2.21
			Average	1.95

Iterations	Statement	Scalar	Vector	Ratio
1000	⍎4⍕R20	.056	.053	1.06
1000	⍎4⍕R50	.110	.074	1.49
500	⍎4⍕R100	.194	.112	1.73
500	⍎4⍕R200	.366	.186	1.97
200	⍎4⍕R400	.715	.350	2.04
200	⍎4⍕R800	1.390	.670	2.07
100	⍎4⍕R1600	2.920	1.410	2.07
100	⍎4⍕R3200	5.700	2.730	2.09
100	⍎4⍕R6400	11.270	5.310	2.12
100	⍎4⍕R12800	22.350	10.470	2.13
			Average	1.88

Iterations	Statement	Scalar	Vector	Ratio
1000	⍎5⍕R20	.058	.053	1.09
1000	⍎5⍕R50	.108	.073	1.48
500	⍎5⍕R100	.196	.110	1.78
500	⍎5⍕R200	.370	.186	1.99
200	⍎5⍕R400	.720	.345	2.09
200	⍎5⍕R800	1.410	.665	2.12
100	⍎5⍕R1600	2.930	1.410	2.08
100	⍎5⍕R3200	5.750	2.750	2.09
100	⍎5⍕R6400	11.350	5.340	2.13
100	⍎5⍕R12800	22.580	10.510	2.15
			Average	1.90

External Function ATR (Array To Record)

Iterations	Statement	Scalar	Vector	Ratio
100	I2 ATR I100	.220	.200	1.10
100	I2 ATR I200	.240	.210	1.14
50	I2 ATR I400	.300	.240	1.25
50	I2 ATR I800	.380	.280	1.36
25	I2 ATR I1600	.600	.320	1.88
25	I2 ATR I3200	.960	.480	2.00
10	I2 ATR I6400	1.900	.900	2.11
10	I2 ATR I12800	3.600	1.600	2.25
			Average	1.64

Iterations	Statement	Scalar	Vector	Ratio
100	E16 ATR I100	.230	.200	1.15
100	E16 ATR I200	.300	.220	1.36
50	E16 ATR I400	.400	.280	1.43
50	E16 ATR I800	.700	.500	1.40
25	E16 ATR I1600	1.160	.720	1.61
25	E16 ATR I3200	2.080	1.200	1.73
10	E16 ATR I6400	3.900	2.200	1.77
10	E16 ATR I12800	7.400	3.900	1.90
			Average	1.54

Iterations	Statement	Scalar	Vector	Ratio
100	J8 ATR I100	.240	.200	1.20
100	J8 ATR I200	.260	.210	1.24
50	J8 ATR I400	.360	.260	1.38
50	J8 ATR I800	.540	.300	1.80
25	J8 ATR I1600	1.000	.560	1.79
25	J8 ATR I3200	1.720	.800	2.15
10	J8 ATR I6400	3.300	1.500	2.20
10	J8 ATR I12800	6.300	2.700	2.33
			Average	1.76

Iterations	Statement	Scalar	Vector	Ratio
100	J32 ATR I100	.240	.220	1.09
100	J32 ATR I200	.460	.230	2.00
50	J32 ATR I400	.600	.500	1.20
50	J32 ATR I800	.920	.700	1.31
25	J32 ATR I1600	1.560	1.160	1.34
25	J32 ATR I3200	2.960	2.080	1.42
10	J32 ATR I6400	5.600	3.900	1.44
10	J32 ATR I12800	10.800	7.300	1.48
			Average	1.41

Iterations	Statement	Scalar	Vector	Ratio
100	E16 ATR R100	.210	.210	1.00
100	E16 ATR R200	.220	.210	1.05
50	E16 ATR R400	.380	.280	1.36
50	E16 ATR R800	.560	.480	1.17
25	E16 ATR R1600	.840	.680	1.24
25	E16 ATR R3200	1.480	1.120	1.32
10	E16 ATR R6400	2.800	2.100	1.33
10	E16 ATR R12800	5.200	3.700	1.41
			Average	1.23

Iterations	Statement	Scalar	Vector	Ratio
100	J32 ATR C100	.220	.220	1.00
100	J32 ATR C200	.370	.220	1.68
50	J32 ATR C400	.540	.500	1.08
50	J32 ATR C800	.800	.720	1.11
25	J32 ATR C1600	1.360	1.200	1.13
25	J32 ATR C3200	2.600	2.160	1.20
10	J32 ATR C6400	4.900	4.100	1.20
10	J32 ATR C12800	9.200	7.700	1.19
			Average	1.20

Iterations	Statement	Scalar	Vector	Ratio
100	E4 ATR B100M	.220	.200	1.10
100	E4 ATR B200M	.240	.210	1.14
50	E4 ATR B400M	.300	.240	1.25
50	E4 ATR B800M	.420	.260	1.62
25	E4 ATR B1600M	.640	.280	2.29
25	E4 ATR B3200M	1.240	.560	2.21
10	E4 ATR B6400M	2.200	.800	2.75
10	E4 ATR B12800M	4.200	1.400	3.00
			Average	1.92

Iterations	Statement	Scalar	Vector	Ratio
100	E16 ATR B100M	.230	.210	1.10
100	E16 ATR B200M	.280	.220	1.27
50	E16 ATR B400M	.360	.280	1.29
50	E16 ATR B800M	.640	.480	1.33
25	E16 ATR B1600M	1.000	.680	1.47
25	E16 ATR B3200M	1.800	1.120	1.61
10	E16 ATR B6400M	3.300	2.000	1.65
10	E16 ATR B12800M	6.200	3.600	1.72
			Average	1.43

Iterations	Statement	Scalar	Vector	Ratio
100	J8 ATR B100M	.230	.210	1.10
100	J8 ATR B200M	.260	.210	1.24
50	J8 ATR B400M	.340	.240	1.42
50	J8 ATR B800M	.480	.280	1.71
25	J8 ATR B1600M	.920	.480	1.92
25	J8 ATR B3200M	1.520	.680	2.24
10	J8 ATR B6400M	2.800	1.100	2.55
10	J8 ATR B12800M	5.300	2.000	2.65
			Average	1.85

Iterations	Statement	Scalar	Vector	Ratio
100	J32 ATR B100M	.230	.220	1.05
100	J32 ATR B200M	.380	.220	1.73
50	J32 ATR B400M	.580	.500	1.16
50	J32 ATR B800M	.880	.700	1.26
25	J32 ATR B1600M	1.480	1.160	1.28
25	J32 ATR B3200M	2.760	2.000	1.38
10	J32 ATR B6400M	5.300	3.800	1.39
10	J32 ATR B12800M	10.000	7.100	1.41
			Average	1.33

Iterations	Statement	Scalar	Vector	Ratio
100	J32 ATR R100	.220	.220	1.00
100	J32 ATR R200	.370	.230	1.61
50	J32 ATR R400	.540	.500	1.08
50	J32 ATR R800	.780	.780	1.00
25	J32 ATR R1600	1.320	1.120	1.18
25	J32 ATR R3200	2.440	2.040	1.20
10	J32 ATR R6400	4.500	3.900	1.15
10	J32 ATR R12800	8.600	7.100	1.21
			Average	1.18

Iterations	Statement	Scalar	Vector	Ratio
100	E4 ATR J100	.200	.200	1.00
100	E4 ATR J200	.210	.210	1.00
50	E4 ATR J400	.240	.240	1.00
50	E4 ATR J800	.300	.280	1.07
25	E4 ATR J1600	.400	.360	1.11
25	E4 ATR J3200	.760	.640	1.19
10	E4 ATR J6400	1.300	1.000	1.30
10	E4 ATR J12800	2.200	1.800	1.22
			Average	1.11

Iterations	Statement	Scalar	Vector	Ratio
100	E16 ATR J100	.220	.210	1.05
100	E16 ATR J200	.260	.220	1.18
50	E16 ATR J400	.300	.280	1.07
50	E16 ATR J800	.560	.500	1.12
25	E16 ATR J1600	.840	.720	1.17
25	E16 ATR J3200	1.440	1.200	1.20
10	E16 ATR J6400	2.700	2.200	1.23
10	E16 ATR J12800	5.000	4.000	1.25
			Average	1.16

Iterations	Statement	Scalar	Vector	Ratio
100	J8 ATR J100	.210	.210	1.00
100	J8 ATR J200	.230	.220	1.05
50	J8 ATR J400	.300	.260	1.15
50	J8 ATR J800	.400	.320	1.25
25	J8 ATR J1600	.680	.560	1.21
25	J8 ATR J3200	1.160	.920	1.26
10	J8 ATR J6400	2.000	1.500	1.33
10	J8 ATR J12800	3.800	2.700	1.41
			Average	1.21

Iterations	Statement	Scalar	Vector	Ratio
100	J32 ATR J100	.240	.240	1.00
100	J32 ATR J200	.460	.260	1.77
50	J32 ATR J400	.640	.600	1.07
50	J32 ATR J800	.940	.860	1.09
25	J32 ATR J1600	1.600	1.480	1.08
25	J32 ATR J3200	2.800	2.560	1.09
10	J32 ATR J6400	5.200	4.900	1.06
10	J32 ATR J12800	9.800	9.000	1.09
			Average	1.16

Strings of Scalar Functions

Iterations	Statement	Scalar	Vector	Ratio
200	3.1-A1+B1+2.1×A1÷1.12-A1⌊B1×3.1	.295	.125	2.36
100	3.1-A2+B2+2.1×A2÷1.12-A2⌊B2×3.1	.450	.160	2.81
50	3.1-A3+B3+2.1×A3÷1.12-A3⌊B3×3.1	.940	.260	3.77
50	3.1-A4+B4+2.1×A4÷1.12-A4⌊B4×3.1	1.760	.420	4.19
25	3.1-A5+B5+2.1×A5÷1.12-A5⌊B5×3.1	3.920	.840	4.67
25	3.1-A6+B6+2.1×A6÷1.12-A6⌊B6×3.1	7.640	1.600	4.78
25	3.1-A7+B7+2.1×A7÷1.12-A7⌊B7×3.1	15.800	3.000	5.27
15	3.1-A8+B8+2.1×A8÷1.12-A8⌊B8×3.1	31.267	5.867	5.23
			Average	4.15

Iterations	Statement	Scalar	Vector	Ratio
200	3.1-(A1+(B1+(2.1×(A1÷(1.12-(A1⌊(B1×3.1)))))))	.320	.250	1.28
100	3.1-(A2+(B2+(2.1×(A2÷(1.12-(A2⌊(B2×3.1)))))))	.470	.290	1.62
50	3.1-(A3+(B3+(2.1×(A3÷(1.12-(A3⌊(B3×3.1)))))))	1.100	.560	1.96
50	3.1-(A4+(B4+(2.1×(A4÷(1.12-(A4⌊(B4×3.1)))))))	1.780	.920	1.93
25	3.1-(A5+(B5+(2.1×(A5÷(1.12-(A5⌊(B5×3.1)))))))	3.960	2.120	1.87
25	3.1-(A6+(B6+(2.1×(A6÷(1.12-(A6⌊(B6×3.1)))))))	7.600	3.960	1.92
25	3.1-(A7+(B7+(2.1×(A7÷(1.12-(A7⌊(B7×3.1)))))))	15.640	7.880	1.99
15	3.1-(A8+(B8+(2.1×(A8÷(1.12-(A8⌊(B8×3.1)))))))	31.200	15.200	2.05
			Average	1.83

For additional information, refer to "Strings of Scalar Functions" on page 88.

Appendix A. APL2 Code Used in Timing Analysis

Following are the data, the timing functions, and approach used in statement creation for performing the timings.

Data

The following functions create the global variables used for the statement being timed.

The prefix "*I*" implies integer, the prefix "*R*" real, the prefix "*C*" complex and the prefix "*B*" implies boolean.

A vector has no underbars in its name. For example, *R*100 implies a vector of real data with a length of 100. *R*100_100 implies a two-dimensional array with a length of 100 in each dimension.

```
     ∇
[0]    BVECTOR;⎕IO;M
[1]    ⎕IO←0
[2]    ⎕RL←7*5
[3]    M←2
[4]    B100O←100ρ1
[5]    B200O←200ρ1
[6]    B400O←400ρ1
[7]    B800O←800ρ1
[8]    B1600O←1600ρ1
[9]    B3200O←3200ρ1
[10]   B6400O←6400ρ1
[11]   B12800O12800ρ1
[12]   B100Z←100ρ0
[13]   B200Z←200ρ0
[14]   B400Z←400ρ0
[15]   B800Z←800ρ0
[16]   B1600Z←1600ρ0
[17]   B3200Z←3200ρ0
[18]   B6400Z←6400ρ0
[19]   B12800Z←12800ρ0
[20]   B20M←1=?20ρM
[21]   B50M←1=?50ρM
[22]   B100M←1=?100ρM
[23]   B200M←1=?200ρM
[24]   B400M←1=?400ρM
[25]   B800M←1=?800ρM
[26]   B1600M←1=?1600ρM
[27]   B3200M←1=?3200ρM
[28]   B6400M←1=?6400ρM
[29]   B12800M←1=?12800ρM
[30]   B20MM←1=?20ρM
[31]   B50MM←1=?50ρM
[32]   B100MM←1=?100ρM
[33]   B200MM←1=?200ρM
[34]   B400MM←1=?400ρM
[35]   B800MM←1=?800ρM
[36]   B1600MM←1=?1600ρM
[37]   B3200MM←1=?3200ρM
[38]   B6400MM←1=?6400ρM
[39]   B12800MM←1=?12800ρM
     ∇ 02/02/1989 15.59.16 (GMT-5)

     ∇
[0]    CMATRIX;Z
[1]    Z←COMPLEX 12800
[2]    C100_100←100 100ρZ
[3]    C200_200←200 200ρZ
[4]    C400_400←400 400ρZ
     ∇ 08/25/1988  9.15.44 (GMT-5)
```

```
    ∇
[0]   CVECTOR;Z
[1]   Z←COMPLEX 12800
[2]   C100←100↑Z
[3]   C200←200↑Z
[4]   C400←400↑Z
[5]   C800←800↑Z
[6]   C1600←1600↑Z
[7]   C3200←3200↑Z
[8]   C6400←6400↑Z
[9]   C12800←Z
    ∇ 08/25/1988  9.01.34 (GMT-5)

    ∇
[0]   CVECTORD;Z
[1]   CVECTOR
[2]   Z←COMPLEX 12800
[3]   C100D←¯100↑Z
[4]   C200D←¯200↑Z
[5]   C400D←¯400↑Z
[6]   C800D←¯800↑Z
[7]   C1600D←¯1600↑Z
[8]   C3200D←¯3200↑Z
[9]   C6400D←¯6400↑Z
[10]  C12800D←⌽Z
    ∇ 08/25/1988  9.55.37 (GMT-5)

    ∇
[0]   CVECTORP;Z
[1]   Z←(2*7)×(COMPLEX 12800)×÷¯1×2*31
[2]   C100←100↑Z
[3]   C200←200↑Z
[4]   C400←400↑Z
[5]   C800←800↑Z
[6]   C1600←1600↑Z
[7]   C3200←3200↑Z
[8]   C6400←6400↑Z
[9]   C12800←Z
    ∇ 03/25/1988  9.57.27 (GMT-5)
```

```
      ∇
[0]     C3ARRAY;M
[1]     M←COMPLEX 12800
[2]     C10_2_2←10 2 2⍴M
[3]     C100_2_2←100 2 2⍴M
[4]     C500_2_2←500 2 2⍴M
[5]     C2_10_2←2 10 2⍴M
[6]     C2_100_2←2 100 2⍴M
[7]     C2_500_2←2 500 2⍴M
[8]     C2_2_10←2 2 10⍴M
[9]     C2_2_100←2 2 100⍴M
[10]    C2_2_500←2 2 500⍴M
[11]    C10_5_5←10 5 5⍴M
[12]    C100_5_5←100 5 5⍴M
[13]    C500_5_5←500 5 5⍴M
[14]    C5_10_5←5 10 5⍴M
[15]    C5_100_5←5 100 5⍴M
[16]    C5_500_5←5 500 5⍴M
[17]    C5_5_10←5 5 10⍴M
[18]    C5_5_100←5 5 100⍴M
[19]    C5_5_500←5 5 500⍴M
[20]    C10_10_10←10 10 10⍴M
[21]    C100_10_10←100 10 10⍴M
[22]    C500_10_10←500 10 10⍴M
[23]    C10_10_10←10 10 10⍴M
[24]    C10_100_10←10 100 10⍴M
[25]    C10_500_10←10 500 10⍴M
[26]    C10_10_100←10 10 100⍴M
[27]    C10_10_500←10 10 500⍴M
[28]    C10_20_20←10 20 20⍴M
[29]    C100_20_20←100 20 20⍴M
[30]    C500_20_20←500 20 20⍴M
[31]    C20_10_20←20 10 20⍴M
[32]    C20_100_20←20 100 20⍴M
[33]    C20_500_20←20 500 20⍴M
[34]    C20_20_10←20 20 10⍴M
[35]    C20_20_100←20 20 100⍴M
[36]    C20_20_500←20 20 500⍴M
      ∇ 02/15/1988 15.29.33 (GMT-5)

      ∇
[0]     IMATRIX;M
[1]     M←2*12
[2]     ⎕RL←7*5
[3]     I100_100←?100 100⍴M
[4]     I200_200←?200 200⍴M
[5]     I400_400←?400 400⍴M
      ∇ 02/02/1989 15.54.46 (GMT-5)
      ∇
[0]    IVECTOR;M
[1]    M←2*15
[2]    ⎕RL←7*5
[3]    I100←?100⍴M
[4]    I200←?200⍴M
[5]    I400←?400⍴M
[6]    I800←?800⍴M
[7]    I1600←?1600⍴M
[8]    I3200←?3200⍴M
[9]    I6400←?6400⍴M
[10]   I12800←?12800⍴M
      ∇ 02/02/1989 15.51.01 (GMT-5)

      ∇
[0]    IVECTORD;M
[1]    M←2*15
[2]    IVECTOR
[3]    I100D←?100⍴M
[4]    I200D←?200⍴M
[5]    I400D←?400⍴M
[6]    I800D←?800⍴M
[7]    I1600D←?1600⍴M
[8]    I3200D←?3200⍴M
[9]    I6400D←?6400⍴M
[10]   I12800D←?12800⍴M
      ∇ 02/02/1989 15.49.31 (GMT-5)

      ∇
[0]    IVECTORP;M;⎕RL
[1]    M←2*7   ⍝ Range must be within domain of '*' primitive
[2]    ⎕RL←7*5
[3]    I100←?100⍴M
[4]    I200←?200⍴M
[5]    I400←?400⍴M
[6]    I800←?800⍴M
[7]    I1600←?1600⍴M
[8]    I3200←?3200⍴M
[9]    I6400←?6400⍴M
[10]   I12800←?12800⍴M
      ∇ 01/25/1989 15.30.41 (GMT-5)
```

```
     ∇
[0]    I3ARRAY;M;⎕RL
[1]    M←2*15
[2]    ⎕RL←7*5
[3]    I10_2_2←?10 2 2⍴M
[4]    I100_2_2←?100 2 2⍴M
[5]    I500_2_2←?500 2 2⍴M
[6]    I2_10_2←?2 10 2⍴M
[7]    I2_100_2←?2 100 2⍴M
[8]    I2_500_2←?2 500 2⍴M
[9]    I2_2_10←?2 2 10⍴M
[10]   I2_2_100←?2 2 100⍴M
[11]   I2_2_500←?2 2 500⍴M
[12]   I10_5_5←?10 5 5⍴M
[13]   I100_5_5←?100 5 5⍴M
[14]   I500_5_5←?500 5 5⍴M
[15]   I5_10_5←?5 10 5⍴M
[16]   I5_100_5←?5 100 5⍴M
[17]   I5_500_5←?5 500 5⍴M
[18]   I5_5_10←?5 5 10⍴M
[19]   I5_5_100←?5 5 100⍴M
[20]   I5_5_500←?5 5 500⍴M
[21]   I10_10_10←?10 10 10⍴M
[22]   I100_10_10←?100 10 10⍴M
[23]   I500_10_10←?500 10 10⍴M
[24]   I10_10_10←?10 10 10⍴M
[25]   I10_100_10←?10 100 10⍴M
[26]   I10_500_10←?10 500 10⍴M
[27]   I10_10_100←?10 10 100⍴M
[28]   I10_10_500←?10 10 500⍴M
[29]   I10_20_20←?10 20 20⍴M
[30]   I100_20_20←?100 20 20⍴M
[31]   I500_20_20←?500 20 20⍴M
[32]   I20_10_20←?20 10 20⍴M
[33]   I20_100_20←?20 100 20⍴M
[34]   I20_500_20←?20 500 20⍴M
[35]   I20_20_10←?20 20 10⍴M
[36]   I20_20_100←?20 20 100⍴M
[37]   I20_20_500←?20 20 500⍴M
     ∇ 01/25/1989 15.32.22 (GMT-5)
     ∇
[0]    PVECTOR;M;⎕RL;L
[1]    M←2*6
[2]    L←2*31
[3]    ⎕RL←7*5
[4]    I100←?100ρM
[5]    I200←?200ρM
[6]    I400←?400ρM
[7]    I800←?800ρM
[8]    I1600←?1600ρM
[9]    I3200←?3200ρM
[10]   I6400←?6400ρM
[11]   I12800←?12800ρM
[12]   R100←.1×?100ρL
[13]   R200←.1×?200ρL
[14]   R400←.1×?400ρL
[15]   R800←.1×?800ρL
[16]   R1600←.1×?1600ρL
[17]   R3200←.1×?3200ρL
[18]   R6400←.1×?6400ρL
[19]   R12800←.1×?12800ρL
[20]   PI100←?100ρ5
[21]   PI200←?200ρ5
[22]   PI400←?400ρ5
[23]   PI800←?800ρ5
[24]   PI1600←?1600ρ5
[25]   PI3200←?3200ρ5
[26]   PI6400←?6400ρ5
[27]   PI12800←?12800ρ5
[28]   PR100←.1×PI100
[29]   PR200←.1×PI200
[30]   PR400←.1×PI400
[31]   PR800←.1×PI800
[32]   PR1600←.1×PI1600
[33]   PR3200←.1×?PI3200
[34]   PR6400←.1×?PI6400
[35]   PR12800←.1×?PI12800
     ∇ 01/31/1989 14.21.33 (GMT-5)

     ∇
[0]    RADIANS;M;⎕IO
[1]   ⍝ Used by the CIRCULAR statements where applicable
[2]    ⎕IO←0
[3]    M←○(⍳12801)×÷180     ⍝ Generate radians for degrees
[4]    R100←100↑M
[5]    R200←200↑M
[6]    R400←400↑M
[7]    R800←800↑M
[8]    R1600←1600↑M
[9]    R3200←3200↑M
[10]   R6400←6400↑M
[11]   R12800←M
     ∇ 01/25/1989 15.33.35 (GMT-5)
     ∇
[0]    RMATRIX;M;L
[1]    ⎕RL←7*5
[2]    M←¯1+2*31
[3]    L←2*32
[4]    R20_20←L×(?20 20⍴M)÷M
[5]    R50_50←L×(?50 50⍴M)÷M
[6]    R100_100←L×(?100 100⍴M)÷M
[7]    R200_200←L×(?200 200⍴M)÷M
[8]    R400_400←L×(?400 400⍴M)÷M
     ∇ 02/02/1989 15.55.01 (GMT-5)

     ∇
[0]    RVECTOR;M;⎕IO
[1]   ⍝ Create real vectors
[2]    ⎕IO←0
[3]    ⎕RL←7*5
[4]    M←¯1+2*31
[5]    R20←.1×?20⍴M
[6]    R50←.1×?50⍴M
[7]    R100←.1×?100⍴M
[8]    R200←.1×?200⍴M
[9]    R400←.1×?400⍴M
[10]   R800←.1×?800⍴M
[11]   R1600←.1×?1600⍴M
[12]   R3200←.1×?3200⍴M
[13]   R6400←.1×?6400⍴M
[14]   R12800←.1×?12800⍴M
     ∇ 02/02/1989 15.52.37 (GMT-5)

     ∇
[0]    RVECTORD;M;⎕IO
[1]    ⎕IO←0
[2]    M←¯1+2*31
[3]    RVECTOR
[4]    R20D←.1×?20⍴M
[5]    R50D←.1×?50⍴M
[6]    R100D←.1×?100⍴M
[7]    R200D←.1×?200⍴M
[8]    R400D←.1×?400⍴M
[9]    R800D←.1×?800⍴M
[10]   R1600D←.1×?1600⍴M
[11]   R3200D←.1×?3200⍴M
[12]   R6400D←.1×?6400⍴M
[13]   R12800D←.1×?12800⍴M
     ∇ 01/26/1989  8.51.07 (GMT-5)
     ∇
[0]    RVECTORP;M
[1]    ⎕RL←7*5
[2]    M←2*7
[3]    R100←.1×?100ρM
[4]    R200←.1×?200ρM
[5]    R400←.1×?400ρM
[6]    R800←.1×?800ρM
[7]    R1600←.1×?1600ρM
[8]    R3200←.1×?3200ρM
[9]    R6400←.1×?6400ρM
[10]   R12800←.1×?12800ρM
     ∇ 02/02/1989 16.00.35 (GMT-5)
     ∇
[0]    R3ARRAY;M;L;⎕RL
[1]    ⎕RL←7*5
[2]    M←¯1+2*31
[3]    L←2*32
[4]    R10_2_2←L×(?10 2 2⍴M)+M
[5]    R100_2_2←L×(?100 2 2⍴M)+M
[6]    R500_2_2←L×(?500 2 2⍴M)+M
[7]    R2_10_2←L×(?2 10 2⍴M)+M
[8]    R2_100_2←L×(?2 100 2⍴M)+M
[9]    R2_500_2←L×(?2 500 2⍴M)+M
[10]   R2_2_10←L×(?2 2 10⍴M)+M
[11]   R2_2_100←L×(?2 2 100⍴M)+M
[12]   R2_2_500←L×(?2 2 500⍴M)+M
[13]   R10_5_5←L×(?10 5 5⍴M)+M
[14]   R100_5_5←L×(?100 5 5⍴M)+M
[15]   R500_5_5←L×(?500 5 5⍴M)+M
[16]   R5_10_5←L×(?5 10 5⍴M)+M
[17]   R5_100_5←L×(?5 100 5⍴M)+M
[18]   R5_500_5←L×(?5 500 5⍴M)+M
[19]   R5_5_10←L×(?5 5 10⍴M)+M
[20]   R5_5_100←L×(?5 5 100⍴M)+M
[21]   R5_5_500←L×(?5 5 500⍴M)+M
[22]   R10_10_10←L×(?10 10 10⍴M)+M
[23]   R100_10_10←L×(?100 10 10⍴M)+M
[24]   R500_10_10←L×(?500 10 10⍴M)+M
[25]   R10_10_10←L×(?10 10 10⍴M)+M
[26]   R10_100_10←L×(?10 100 10⍴M)+M
[27]   R10_500_10←L×(?10 500 10⍴M)+M
[28]   R10_10_100←L×(?10 10 100⍴M)+M
[29]   R10_10_500←L×(?10 10 500⍴M)+M
[30]   R10_20_20←L×(?10 20 20⍴M)+M
[31]   R100_20_20←L×(?100 20 20⍴M)+M
[32]   R500_20_20←L×(?500 20 20⍴M)+M
[33]   R20_10_20←L×(?20 10 20⍴M)+M
[34]   R20_100_20←L×(?20 100 20⍴M)+M
[35]   R20_500_20←L×(?20 500 20⍴M)+M
[36]   R20_20_10←L×(?20 20 10⍴M)+M
[37]   R20_20_100←L×(?20 20 100⍴M)+M
[38]   R20_20_500←L×(?20 20 500⍴M)+M
     ∇ 01/25/1989 15.38.59 (GMT-5)
```

```
    ∇
[0]   SVECTOR;M;⎕IO;⎕RL
[1]  ⍝ Special data for the  0○R   R must be between 0-1
[2]   ⎕RL←7*5
[3]   ⎕IO←0
[4]   M←¯1+2*31
[5]   R100←(?100⍴M)÷M
[6]   R200←(?200⍴M)÷M
[7]   R400←(?400⍴M)÷M
[8]   R800←(?800⍴M)÷M
[9]   R1600←(?1600⍴M)÷M
[10]  R3200←(?3200⍴M)÷M
[11]  R6400←(?6400⍴M)÷M
[12]  R12800←(?12800⍴M)÷M
    ∇ 01/25/1989 15.40.24 (GMT-5)

    ∇
[0]   STRINGV;M;⎕IO;⎕RL;L
[1]   ⎕IO←0
[2]   ⎕RL←7*5
[3]   M←¯1+2*31
[4]   L←2*32
[5]   A1←L×(?100⍴M)÷M    ⍝ Generate data for string operations
[6]   A2←L×(?200⍴M)÷M    ⍝ Done this way to avoid truncation
[7]   A3←L×(?400⍴M)÷M    ⍝ of statement in report
[8]   A4←L×(?800⍴M)÷M
[9]   A5←L×(?1600⍴M)÷M
[10]  A6←L×(?3200⍴M)÷M
[11]  A7←L×(?6400⍴M)÷M
[12]  A8←L×(?12800⍴M)÷M
[13]  B1←L×(?100⍴M)÷M
[14]  B2←L×(?200⍴M)÷M
[15]  B3←L×(?400⍴M)÷M
[16]  B4←L×(?800⍴M)÷M
[17]  B5←L×(?1600⍴M)÷M
[18]  B6←L×(?3200⍴M)÷M
[19]  B7←L×(?6400⍴M)÷M
[20]  B8←L×(?12800⍴M)÷M
    ∇ 01/11/1988 10.39.43 (GMT-5)
```

Timing Functions

The following functions are general timing functions (as opposed to functions in other sections which were developed exclusively for this technical report).

```
     ∇
[0]    R←EXECUTE X;T;FN;Z;CODE;N
[1]   ⍝ Function fixes statement "X" as a function and executes same
[2]   ⍝ Number of iterations is 1⊃X ; Statement to be timed is  2⊃X
[3]    N←1⊃X                    ⍝ Hold number of iterations
[4]    FN←⊂'Z←CODE;T'           ⍝ Define a function name for ⎕FX
[5]    FN←FN,⊂'T←GCOL'          ⍝ Force garbage collect as 1st line of fn
[6]    FN←FN,⊂'T←⎕AI[2]'        ⍝ Make 2nd statement start of CPU timer
[7]    FN←FN,(⊂'Z←'),¨N⍴X[2]
[8]    FN←FN,⊂'Z←⎕AI[2]-T'      ⍝ Hold total execution time
[9]    Z←⎕FX FN                 ⍝ Fix function
[10]   Z←CODE                   ⍝ Execute statement "X" as a function
[11]   R←Z÷N                    ⍝ Return average ⎕AI[2] CPU time
     ∇ 04/12/1988 16.02.55 (GMT-5)


     ∇
[0]    X REPORT R;RATIO;VON;VOFF;H;AVG;Z;RZ
[1]   ⍝ Function generates a REPORT with the statement executed and
[2]   ⍝ the time (in milliseconds) to execute said statement.
[3]   ⍝ With vector off and vector on and resulting ratio VOFF÷VON
[4]   ⍝ Left argument is the statements that were timed, # of iterations
[5]   ⍝ NOTE:  statements over 21 characters in length will be truncated
[6]   ⍝ 1st item of right arg is vec OFF times, 2nd item is vec ON times
[7]    X←(5 0⍕X[;,1]),(((↑⍴X),7)⍴' '),⍕X[;,2]
[8]    RATIO←RZ←(1⊃R)÷2⊃R
[9]    AVG←Z←(+/RATIO)÷⍴,RATIO
[10]   VOFF←10 3⍕,[1.5]1⊃R                 ⍝ 1st a matrix then round
[11]   VON←10 3⍕,[1.5]2⊃R
[12]   RATIO←8 2⍕,[1.5]RATIO
[13]   AVG←5 2⍕AVG
[14]   'Iterations   Statement',(16⍴' '),'Scalar    Vector   Ratio'
[15]   (((↑⍴X),34)↑X),VOFF,VON,RATIO
[16]   ¯62↑'Average',AVG
[17]   H←(8 1⍕⌊/RZ),('  -'),(5 1⍕⌈/RZ),10 1⍕Z
[18]   SUMRY←SUMRY,[1](¯1↑⍴SUMRY)↑STATEMENT,H
     ∇ 04/12/1988 16.02.55 (GMT-5)


     ∇
[0]    RUN X;J;Y;VOFF;VON
[1]   ⍝ Function sets up the call for the execution of the right arg "X"
[2]    J←VECTOR 1              ⍝ Turn vector off using locked function
[3]   ⍝                          see WS UTILS for the unlocked version
[4]    Y←SETUP X
[5]    VOFF←EXECUTE¨Y          ⍝ Create the statement into a function
[6]    J←VECTOR 2              ⍝ Turn vector on using locked function
[7]    VON←EXECUTE¨Y           ⍝ Redo with vector on
[8]    (⊃[2]Y)REPORT VOFF VON
[9]    J←⎕EX¨VECTORVARS VECTORDVARS MATRIXVARS MAPVARS BOOLEANS VPATTERNS
     ∇ 10/05/1988 14.15.37 (GMT-5)


     ∇
[0]    Z←SETUP FN;CR;C
[1]   ⍝ Right argument is name of function with 1st statement NAME of
[2]   ⍝ function that creates the variables used by statements timed
[3]   ⍝ Function returns the statements and executes the 1st statement
[4]    CR←⎕CR FN              ⍝ Get canonical representation
[5]    CR←1 0↓CR              ⍝ Drop name of function
[6]    ⍎CR[1;]                ⍝ Create variables used in timing statements
[7]    STATEMENT←25↑,CR[2;]     ⍝ Hold statement type being timed
[8]    Z←⍎¨⊂[2]2 0↓CR           ⍝ Return iterations/statements to be timed
     ∇ 03/14/1988 13.48.29 (GMT-5)
     ∇
[0]    START X;Z;STATEMENT;OPTION;SUMRY
[1]   ⍝ "X" is a character matrix with each row the name of a variable
[2]   ⍝ that has the names of the functions that consists of the number
[3]   ⍝ of iterations and the statement to be timed
[4]   ⍝ To invoke simply enter as eg. START 'MONADIC_SCALAR_PRIMITIVES'
[5]   ⍝ To use garbage collect GCOL function must be privileged
[6]    Z←3 11 ⎕NA 'ATR'
[7]    Z←3 11 ⎕NA 'OPTION'
[8]    Z←'128' OPTION 'SYSDEBUG' ⍝ Privilege bit;for garbage collect
[9]    SUMRY←,[0.5]70↑(31⍴' '),'Scalar+Vector Ratios'
[10]   SUMRY←SUMRY,[1]70↑'Statement',(24⍴' '),'Range        Average'
[11]   X←(¯2↑1 1,⍴X)⍴X
[12]   Z←⊃,/⊂[2]¨⍎¨⊂[2]X   ⍝ Create a names list of each timing function
[13]   RUN¨Z
[14]   Z←'¯128' OPTION 'SYSDEBUG'  ⍝ Restore privilege bit
[15]   Z←⎕EX 'ATR'
[16]   SUMRY
     ∇ 01/26/1989 12.24.49 (GMT-5)

     ∇
[0]    R←VECTOR N
[1]    ⍝ Turn vector on/off under program control
[2]    ⍝ The technique of turning the IBM 3090 Vector Facility
[3]    ⍝ on/off under program control as shown in this function
[4]    ⍝ is subject to change and any direct use of this function
[5]    ⍝ is solely the responsibility of the user.
[6]    ⍝ 25 ⎕IB 1 0  queries vector, returns 1 1 if vector off, 2 2 if on
[7]    ⍝ 25 ⎕IB 1 1  set vector off, returns previous setting, new setting
[8]    ⍝ 25 ⎕IB 1 2  set vector on, returns previous setting, new setting
[9]    ⍝ Executed as locked function, VECTOR 1 sets vector off
[10]   ⍝ Executed as locked function, VECTOR 2 sets vector on
[11]    R←25 ⎕IB 1 N
     ∇ 10/25/1988 14.13.26 (GMT-5)
```

Statement Creation

The following functions are representative of the method used to define the expression and the number of iterations being timed. They can be any executable APL expression. That is, they need not be a simple primitive. See the string ratios as an example of the timing of a multiple number of primitives.

The first statement executed is the name of the function that creates the global variables used during execution.

The second statement is used to represent the general expression that will be timed. It appears under the statement heading in the summary results.

In the subsequent statements the first item is the number of iterations and the second item is the expression that is iterated and timed via the function called ***CODE***.

```
      ∇
[0]     ADDC
[1]     CVECTORD
[2]     CVECTOR+CVECTOR
[3]     1000 'C100+C100D'
[4]     1000 'C200+C200D'
[5]     500 'C400+C400D'
[6]     200 'C800+C800D'
[7]     100 'C1600+C1600D'
[8]     50 'C3200+C3200D'
[9]     50 'C6400+C6400D'
[10]    50 'C12800+C12800D'
      ∇ 01/14/1988 14.17.05 (GMT-5)

      ∇
[0]     ADDI
[1]     IVECTORD
[2]     IVECTOR+IVECTOR
[3]     1000 'I100+I100D'
[4]     1000 'I200+I200D'
[5]     1000 'I400+I400D'
[6]     500 'I800+I800D'
[7]     400 'I1600+I1600D'
[8]     300 'I3200+I3200D'
[9]     200 'I6400+I6400D'
[10]    100 'I12800+I12800D'
      ∇ 01/14/1988 14.14.21 (GMT-5)
      ∇
[0]     ADDR
[1]     RVECTORD
[2]     RVECTOR+RVECTOR
[3]     1000 'R100+R100D'
[4]     1000 'R200+R200D'
[5]     1000 'R400+R400D'
[6]     500 'R800+R800D'
[7]     400 'R1600+R1600D'
[8]     300 'R3200+R3200D'
[9]     200 'R6400+R6400D'
[10]    100 'R12800+R12800D'
      ∇ 01/14/1988 14.14.55 (GMT-5)

      ∇
[0]     PLUSREDMAGI
[1]     IMATRIX
[2]     +/[2]|IMATRIX
[3]     15 '+/[2]|I100_100'
[4]     15 '+/[2]|I200_200'
[5]     15 '+/[2]|I400_400'
      ∇ 12/18/1987  8.34.25 (GMT-5)

      ∇
[0]     PLUSREDMAGR
[1]     RMATRIX
[2]     +/[2]|RMATRIX
[3]     15 '+/[2]|R100_100'
[4]     10 '+/[2]|R200_200'
[5]     10 '+/[2]|R400_400'
      ∇ 12/18/1987  8.34.51 (GMT-5)

      ∇
[0]     RECIPROCALC
[1]     CVECTOR
[2]     ÷CVECTOR
[3]     200 '÷C100'
[4]     200 '÷C200'
[5]     100 '÷C400'
[6]     50 '÷C800'
[7]     50 '÷C1600'
[8]     25 '÷C3200'
[9]     25 '÷C6400'
[10]    20 '÷C12800'
      ∇ 01/11/1988 12.06.21 (GMT-5)
      ∇
[0]     RECIPROCALI
[1]     IVECTOR
[2]     ÷IVECTOR
[3]     2000 '÷I100'
[4]     2000 '÷I200'
[5]     1000 '÷I400'
[6]     500 '÷I800'
[7]     400 '÷I1600'
[8]     300 '÷I3200'
[9]     200 '÷I6400'
[10]    100 '÷I12800'
      ∇ 01/14/1988 13.58.56 (GMT-5)

      ∇
[0]     RECIPROCALR
[1]     RVECTOR
[2]     ÷RVECTOR
[3]     2000 '÷R100'
[4]     2000 '÷R200'
[5]     1000 '÷R400'
[6]     500 '÷R800'
[7]     400 '÷R1600'
[8]     300 '÷R3200'
[9]     200 '÷R6400'
[10]    100 '÷R12800'
      ∇ 01/14/1988 14.41.58 (GMT-5)
```

Appendix B. Performance Enhancements

The following performance items are part of APL2 Version 1 Release 3.

Interpreter Performance Enhancements

A number of primitive functions and operators run faster in Release 3, because of improved algorithms. (This speed-up is not related to the Vector Facility.) These functions and operators include:

- **Pick** (*L*⊃*R*)
- **First of *n*-drop** (↑*N*↓*R*)
- **Catenate** (*L*,*R*) with rank greater than one
- **Vector notation**
- **Compress / Replicate** (*LO*/*R*)
- **Expand** (*LO*\R)
- **Membership** (*L*∊*R*) with fullword integer arguments or vector of character vector arguments
- **Index of** (*L*⍳*R*) with fullword integer arguments or vector of character vector arguments
- **Without** (*L*~*R*) with fullword integer arguments or vector of character vector arguments
- Certain cases of **logarithm** (*L*⍟*R*), **exponential** (**R*), **magnitude** (|*R*), **power** (*L***R*), and **circular functions** (*L*○*R*)
- Most primitive scalar functions with scalar arguments and several commonly used scalar functions with depth-1 arguments

Packaged Workspaces

A **packaged workspace** is a new concept in APL2 Release 3. Packaged workspaces can change the way APL2 applications are conceived. The main benefits of this facility are:

- Isolation of name scopes
- Ability to share APL code
- Savings in space and increased overall performance

With APL2 Release 3, workspaces may be "packaged" and converted to load modules. On MVS, these load modules may be placed in the LPA (Link Pack Area), and on VM they may be placed in DCSS (Discontiguous Shared Segment). By doing this, they may be shared on a read-only basis between multiple APL2 users at the same time.

Objects in packaged workspaces are **external functions**. They can be accessed using *⎕NA*, in a similar fashion to calls to FORTRAN or Assembler external routines. Packaged workspaces are loaded dynamically when the objects in them are accessed.

Each packaged workspace contains its own "name scope." That is to say, names in packaged workspaces do not conflict with names in the user's workspace or in other packaged workspaces. Thus, packaged workspaces provide an attractive means by which APL applications may be implemented, combined with other applications, and provided to users.

A number of packaged workspaces are provided with APL2. Among them are:

- *TIME*— a set of functions which allows timing of applications and provides performance information to application developers.
- *PACKAGE*— contains functions that allow saved workspaces to be converted to packaged workspaces.
- *DFMT*, *CMSIVP*, *TSOIVP*— perform various utility functions.

For more information on packaged workspaces, refer to "*APL2 Programming: System Services Reference.*"

Using ESSL Routines

The "*Engineering and Scientific Subroutine Library (5668-863)*" contains high-performance routines that take full advantage of the 3090 Vector Facility. These routines may now be called from APL2 Release 3, using Associated Processor 11. A "names" file for the ESSL routines is provided as part of the ESSL program product (Release 2 or 3).

Because the ESSL library is highly tuned for the 3090 Vector Facility, replacing CPU-intensive code in APL2 by the comparable ESSL routine may result in a performance gain.

Calling ESSL routines is documented in "*APL2 Programming: Using the Supplied Routines.*" The Processor 11 interface is explained in "*APL2 Programming: System Services Reference.*"

Additional References

1. Furois P.C.
   "*Performance Analysis of APL2 Expressions and
   Comparable ESSL Routine Calls Via Processor 11*"
   TR 21.1077
   NIC Center, Kingston Laboratory.

2. McComb J., Kapple D., Schmidt S.
   "*Engineering and Scientific Subroutine Library (ESSL)*"
   Program Number (5668-863)
   TR 21.1058
   Data System Division Kingston Laboratory.

3. McComb J., Kapple D., Schmidt S.
   "*Engineering and Scientific Subroutine Library Release 2
   Performance on the IBM 3090 Vector Facility*"
   TR 21.1059
   Data System Division Kingston Laboratory.

4. Khislavsky L., McComb J., Oakley D., Pandian M., Schmidt S., Su J.
   "*Engineering and Scientific Subroutine Library (ESSL) Release 3 Performance on the IBM ES/3090 Vector Facility*"
   GG66-0276
   Washington Systems Center Technical Bulletin

Vector Facility Exploitation

When running on an IBM 3090 CPU with the Vector Facility, APL2 automatically recognizes the presence of that facility and uses it where applicable to improve the performance of APL2 applications.

No modifications to existing APL2 applications are required to use these facilities.

A number of the APL2 primitive scalar functions and operators have been recoded to utilize the 3090 Vector Facility, if it is available. These primitives are listed below.

Strings of Scalar Functions

The APL2 syntax analyzer has been extended to recognize strings of scalar functions that can be executed using the 3090 Vector Facility. The interpreter now builds optimized loops to perform the sequence of operations. A string of scalar functions is an expression of a form such as ***A+B×C÷D***. Vectorization will occur if at least one of the variables contains 20 or more elements.

Functions Enabled for the Vector Facility

In APL2 Release 3, the primitive functions, operators, and idioms listed in this section utilize the 3090 Vector Facility if it is available.

Not all cases of these primitives utilize the Vector Facility. In general, these operations execute in vector mode if the arguments are arrays of type real and of sufficient size to use the Vector Facility to advantage. Some cases with integer and complex arguments also execute in vector mode.

Monadic Scalar Functions (Integer, Real or Complex Arguments)

The following monadic scalar functions utilize the 3090 Vector Facility:

- Complex Conjugate .... +
- Negative .... −
- Direction .... ×
- Natural logarithm .... ⍟
- Reciprocal .... ÷
- Exponential .... *
- Pi times .... ○
- Magnitude .... |

Monadic Scalar Functions (Boolean Arguments)

Additional Vector Facilty support in SPE PL34409 includes:

• Negative	−
• Reciprocal	÷
• Exponential	*
• Natural logarithm	⍟
• Pi times	○
• Roll	?

Vectorization occurs if ×/⍴ is at least 20 elements.

Dyadic Scalar Functions (Integer, Real or Complex Arguments)

The following dyadic scalar functions utilize the 3090 Vector Facility:

• Addition	+
• Subtraction	−
• Multiplication	×
• Division	÷
• Minimum	⌊
• Maximum	⌈
• Power	*
• Square root	*.5
• Base 10 logarithm	10⍟
• Relational functions	<, ≤, =, ≥, >, ≠
• Circular functions	¯12, ¯11, ¯10, ¯6, ¯4, 0, 1, 2, 3, 4, 9, 10, 11, 12

Dyadic Scalar Functions (Boolean Arguments)

Additional Vector Facilty support in SPE PL34409 includes:

• Multiplication	×
• Residue	|
• Minimum	⌊
• Maximum	⌈
• Power	*
• Binomial	!
• And	∧
• Nand	⍲
• Or	∨
• Nor	⍱
• Relational functions	<, ≤ = ≥ > ≠

Vectorization occurs if ×/⍴ is at least 640 elements.

Circular Functions (Boolean Arguments)

Additional Vector Facilty support in SPE PL34409 includes:

- Circular functions .... ¯12, ¯11, ¯6, ¯5, ¯4, ¯3, ¯2, ¯1,
  0, 1, 2, 3, 4, 5, 6, 7, 8, 11, 12

Vectorization occurs if ×/⍴ is at least 20 elements.

Other Functions and Operators

The following functions and operators also utilize the 3090 Vector Facility:

- Inner Product .... +.× (SPE PL34409 allows one boolean argument)
- Reduction .... +/*B* (*B* is a Boolean vector)
- Reduction .... +/*R* (*R* is a Real vector)
- Reduction .... ⌈/*R* (*R* is a Real vector)
- Reduction .... ⌊/*R* (*R* is a Real vector)
- Expand .... *B*\V (*B* is a Boolean vector)
- General reduction .... *LO*/[*n*]*R*
- General inner product .... *L LO*.*RO R*
- General outer product .... *L* ∘.*RO R*

When reduction is applied to arrays with two or more dimensions, vectorization depends upon the existence of a vectorized routine for the left operand function, the axis of reduction, and the relative dimensions of the axes. Similar criteria apply to inner and outer product.

New Functions Supported by SPE PL34409

Additional 3090 Vector Facility support is now included for the following functions and operators:

- Catenate .... *B*,*I* *B*,*E* *B*,*Z* *J*,*I* *J*,*E* *J*,*Z* *I*,*E* *I*,*Z* *E*,*Z*

  (B = Boolean, I = Integer, E = Real, Z = Complex, J = Integer Progression)

  These work in either direction, on any rank simple array, with or without axis. Vectorization occurs if the lower type (the one to be converted) is at least 20 elements.

- Reverse .... ⌽

  On integer, integer progression, real and complex. The length of the axis chosen must be at least 20 elements.

- Transpose .... ⍉

  On integer, integer progression, real and complex. The product of the dimensions to the right of the chosen axis must be at least 20 elements.

- Pair-wise Reduction .... 2+/ or 2-/

  Integer and real arrays with at least 21 elements along the axis of reduction. Vectorization occurs only for pair-wise reduction and only for the functions "+" or "-" as operands.

- Indexed Assignment .... *N*[;*I*]←

  Integer and real arrays using at least 20 fullword integer indices in the last dimension. For all but the last dimension, indices must be elided.

- Indexed Reference .... *N*[;*I*]

  Integer and real arrays with fullword integer indices "*I*" in the last dimension as follows:

  - If dimensions other than the last are indexed, (⍴*I*)≥20
  - If all other dimension indices are elided, (⍴⍴*I*)=0 or (⍴*I*)≥4 and the array must contain at least 8 rows.

- Array to Record .... *ATR*

APL2 Idioms

The following APL2 idioms utilize the 3090 Vector Facility:

- *R*⍳⌈/*R*
- *R*⍳⌊/*R*
- +/|
- ⌈/|
- ⌊/|
- ⍎*N*⍕ (1≤*N*≤5, new idiom in SPE PL34409)

Other Improvements in SPE PL34409

Additional performance improvements in SPE PL34409, independent of the Vector Facility, include:

- Idiom recognition for:

  - Conditional execute .... ⍎*expression*/*constant*
  - Round to *n* decimal places .... ⍎*N*⍕

- Partition .... ⊂

- Out ....)*OUT*
- In ....)*IN*

The following apply only to the MVS/TSO environment:

- Copy ....)*COPY*
- Migrate Copy ....)*MCOPY*
- Protect Copy ....)*PCOPY*

)COPY, *)MCOPY* and *)PCOPY* have always needed a good deal of temporary storage to be able to access the active workspace, the source workspace, and intermediate copies of objects being copied. In the past this storage has been allocated

- In virtual storage if available, or
- In the user's private file library (F0) if allocated, or
- In the sequential files identified by CPYSPILL and CPYSWAP.

The private file library is much slower than the sequential files for this purpose, so preference is now given to CPYSPILL and CPYSWAP if allocated. This allows users to allocate an F0 library for session manager log and/or private AP121 files without incurring a performance penalty in *)COPY*.

Support for Very Large Arrays

The TSO SPE removes the 16 megabyte size limit for simple arrays. It also lifts the 128 megabyte limit on active and saved workspace size, except that:

1. The active workspace must reside in a contiguous range of virtual addresses in the primary MVS/XA or MVS/ESA address space. MVS storage allocation imposes an addressability limit of 1008 megabytes. Hence, all programs and data areas (including the APL workspace) must fit within this limit.
2. Workspaces saved in VSAM libraries are still limited to 128 megabytes each. Because of performance characteristics we recommend that very large workspaces be saved in sequential files rather than VSAM libraries.

Supplied Routine Improvements

- ARRAY to RECORD (*ATR*)

  The external function *ATR* now supports the conversion of many data types required when calling external functions written in languages other than APL.

- CHARACTER to NUMERIC (*CTN*)

  The external function *CTN* now supports a matrix argument, returning a matrix result.

- RECORD to ARRAY (*RTA*)

  The external function *RTA* is extended to allow more general patterns describing simple arrays (not arithmetic progressions):

  - Asterisk (⋆)

    An asterisk (⋆) may be used in place of the shape (⍴) specification in a pattern describing a vector. In this case, the number of items is deduced from the data type specification and the length of the right argument:

```
      A←10⍴ 1.1 2.1 3.1
      SE8←'E8 1 *' ATR A      ⍝ Create record

      'E8 1 *' RTA SE8        ⍝ Get vector from record
1.1 2.1 3.1 1.1 2.1 3.1 1.1 2.1 3.1 1.1

      ⍴'E8 1 *' RTA SE8
10
```

An asterisk may also be used as the leading item of the shape (⍴) in a pattern for a higher rank array. The ×/⍴ specification may be elided or be an asterisk (*). In this case, the length of the leftmost axis is deduced from the number of columns given and the length of the right argument:

```
      '* E8 2 * 5' RTA SE8
1.1 2.1 3.1 1.1 2.1
3.1 1.1 2.1 3.1 1.1

      ⍴SE34
3 32

      'E8 2 * 4' RTA ,SE34
2.1 3.1 4.1 5.1
2.1 3.1 4.1 5.1
2.1 3.1 4.1 5.1

      ⍴'E8 2 * 4' RTA ,SE34
3 4
```

Note that for boolean (B1) and byte (B8) data, values are assumed to reside on byte boundaries. When calculating the shape that replaces the asterisk for other types, the right argument record is truncated to the appropriate length.

CMS SPE PL34409 DCSS Size

After installation of this SPE, the APL2 LOADLIB will be larger than one megabyte. If you intend to save APL2 into a shared segment, you will need to increase the size of the APL2 DCSS(s). Here is a sample DCSS declaration:

```
AP2R30S1 NAMESYS SYSSIZE=1152K,
        SYSNAME=AP2R30S1,VSYSADR=IGNORE,VSYSRES=,
        SYSVOL=ssssss,SYSCYL=,SYSSTRT=(ccc,p),
        SYSPGCT=288,SYSPGNM=(1536-1823),
        SYSHRSG=(96,97,98,99,100,101,102,103,
        104,105,106,107,108,109,110,111,112,113)
```

This increase in size allows some extra space for user processors and packages, but installations adding any additional parts should verify that the new segment size is large enough to accommodate their additions.

Note: The APL2AE LOADLIB will continue to fit in the segment size recommended for base Release 3.