APPENDIX A

Fundamentals of Fortran 90

A.1 Array Assignment

Since one of the main purposes of FORTRAN 90 is providing support for vector and parallel hardware, it is quite natural to assume that FORTRAN 90 contains vector and parallel operations. One way in which this is accomplished is the treatment of vectors and arrays as aggregates in the assignment statement. If X and Y are two arrays of the same dimension, then

```
X = Y
```

copies Y into X element by element. In other words, this assignment is equivalent to

```
X(1) = Y(1)
X(2) = Y(2)
...
X(N) = Y(N)
```

Scalar quantities may be mixed with vector quantities using the convention that a scalar is expanded to a vector of the appropriate dimensions before operations are performed. Thus,

```
X = X + 5.0
```

adds the constant 5.0 to every element of array X.

Array assignments in Fortran 90 are viewed as being executed *simultaneously*; that is, the assignment must be treated so that all input operands are fetched before any output values are stored. For instance, consider

```
X = X/X(2)
```

Even though the value of X(2) is changed by this statement, the original value of X(2) is used throughout, so that the result is the same as

```
T= X(2)
X(1) = X(1)/T
X(2) = X(2)/T
...
X(N) = X(N)/T
```

These semantics match the semantics of vector hardware presented earlier.

A.2 Array Sections

Sections of arrays, including individual rows and columns, may be assigned using triplet notation. If A and B are two dimensional arrays whose subscripts range from 1 to 100 in each dimension; then

```
A(1:100,I) = B(J,1:100)
```

assigns the Jth row of B to the Ith column of A.

The range of iteration for vector assignment may be smaller than a whole row or column. The following assignment could be used to assign the first M elements of the Jth row of B to the first M elements of the Ith column of A.

```
A(1:M,I) = B(J,1:M)
```

This statement has the effect of the assignments:

```
A(1,I) = B(J,1)
```

```
A(2,I) = B(J,2)
...
A(M,I) = B(J,M)
```

even though M might contain a value much smaller than the actual upper bound of these arrays.

The term "triplet" implies three components in the iteration range specification. The third component, when it appears, specifies a "stride" for the index vector in that subscript position. For example, the first M elements of the Jth row of B may be assigned to the first M elements of the Ith column of A *in odd subscript positions* with the following

```
A(1:M*2-1:2,I) = B(J,1:M)
```

Triplet notation is also useful in dealing with operations involving shifted sections. The assignment

```
A(I,1:M) = B(1:M,J) + C(I,3:M+2)
```

has the effect

```
A(I,1) = B(1,J) + C(I,3)
A(I,2) = B(2,J) + C(I,4)
...
A(I,M) = B(M,J) + C(I,M+2)
```

A.3 Parallel Loops

FORTRAN 90 also provides a parallel analog to vector operations. The FORALL loop specifies a DO loop whose iterations can be done in any order. For instance, all elements of a vector A may be incremented by 1 with the following:

```
FORALL I = 1, N {A(I) = A(I) + 1}
```

The FORALL loop is equivalent to another construct that has been in common use: the DOALL. A DOALL allows its iterations to be done in any order, so long as the statements within any one iteration are performed in the specified order. FORALLs and DOALLs are the only kind of parallel loop discussed in this book. Another alternative, the DOACROSS, specifies a parallel loop containing synchronization primitives. FORTRAN 90 contains no mechanism for specifying this kind of parallelism.

A.4 Conditional Assignment

The Fortran WHERE statement will permit an array assignment to be controlled by a conditional masking array. For example,

```
WHERE (A .GT. 0.0) A = A + B
```

specifies that the vector sum of A and B be formed, but that stores back to A take place only in positions where A was originally greater than zero. The semantics of this statement require that it behave as if only components corresponding to the locations where the controlling condition is true are involved in the computation.

In the special case of statements such as

```
WHERE (A .NE. 0.0) B = B/A
```

the semantics require that divide checks arising as a result of evaluating the right hand side not affect the behavior of the program—the code must hide the error from the user. In other words, any error side effects that might occur as a result of evaluating the right hand side in positions where the controlling vector is false are ignored.

A.5 Library Functions

Mathematical library functions, such as SQRT and SIN, are extended on an elementwise basis to vectors and arrays. In addition, new intrinsic-functions are provided, such as inner matrix product (DOTPRODUCT) and transpose (TRANSPOSE). The special function SEQ(1,N) returns an index vector from 1 to N. Reduction functions, much like those in APL, are also provided. For example, SUM applied to a vector returns the sum of all elements in that vector.