

Data Analysis Applications Group (DAAG) Update: Dynamic Memory Allocation in Fortran 90/95

Justin Burruss
2004.04.16

Fortran is widely used at DIII-D, second only to IDL

- A recent user survey showed that Fortran is almost as popular as IDL
- Not too long ago, the FORTRAN 77 compilers were replaced with Fortran 90
 - Note that while FORTRAN 77 is in caps, Fortran 90 is not
- But so what? What's to be gained by using Fortran 90?
- Does it do anything new other than break my old Makefiles?
- The answer is that Fortran 90 introduced many new features, one of which is **dynamic memory allocation**.

Fortran 90 introduces dynamic memory allocation

- In Fortran 77, there is no dynamic memory allocation

```
c      declare array & allocate memory for data  
      real:: aminor(258), t_aminor(258)
```

- Fortran 90 allows for dynamic allocation of memory
- Variables can be declared allocatable, then at run time are allocated

```
c      declare array; will allocate memory at run time  
      real, allocatable :: aminor(:), t_aminor(:)
```

Use ALLOCATE() to allocate memory at runtime

- ALLOCATE attempts to allocate memory for an allocatable variable at run time
- Usage is
 ALLOCATE(*variables*, stat=err)
- *variables* is your list of variables
- err will be set to 0 if the allocation was successful, other numbers indicate error
- Usage example
 ALLOCATE(mydata(1:len), stat=err)

Use allocate when you don't know the data size until runtime

- For example, we have a Fortran code that is to read AMINOR from MDSplus
- Under Fortran 90, the procedure is to first find the length of AMINOR, then allocate memory to store the data

```
c      storage for AMINOR will be allocated later  
real, allocatable :: aminor(:), t_aminor(:)
```

```
c      this function finds length of AMINOR  
len=get_signal_length("AMINOR")
```

```
c      now allocate memory to hold AMINOR  
allocate(aminor(1:len), t_aminor(1:len), stat=err)  
if (err .ne. 0) then  
    stop 'Problem allocating memory for AMINOR'  
endif
```