Fortran "in pillole": prima parte

Annamaria Mazzia

Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate

Corso di Metodi Numerici per l'Ingegneria

Introduzione al Fortran

- Un libro consigliato per imparare il Fortran:
  Flavio Sartoretto e Mario Putti
  Introduzione alla Programmazione per Elaborazioni Numeriche
  Edizioni Libreria Progetto Padova 2008
- Il materiale di queste lezioni verrà messo di volta in volta in rete su
  `http://dispense.dmsa.unipd.it`
  andare poi alla voce Mazzia o Gambolati
  e poi sul corso di metodi numerici per la laurea specialistica in Ing. Meccanica e Ing. dell'Energia

Come si scrive e si esegue un programma in FORTRAN?

- si scrive il programma **sorgente** in un file (con un editor di testo e gli si dà un nome con l'estensione .f se lavoriamo in FORTRAN77 o .f90 se lavoriamo in FORTRAN90...)
- il programma deve essere **compilato**, deve cioè essere tradotto in linguaggio macchina
- se il nostro programma chiama altri programmi (per esempio programmi di libreria) allora bisogna collegarli tra loro - **linkarli**
- si ha in questo modo il programma **eseguibile**
- si può eseguire il programma e ottenere i risultati

Per semplicità spiegheremo il FORTRAN77 - ma il passaggio dal FORTRAN77 al FORTRAN90 diventa semplice, per chi vuole, se si sa bene il FORTRAN77.

Il FORTRAN sul proprio computer

Per chi ha un sistema operativo Linux sul proprio computer, - Debian, Ubuntu, etc - ci si può installare un compilatore FORTRAN sul proprio computer molto facilmente - per esempio il gfortran.
Per chi ha Windows, ci sono compilatori free che si possono scaricare dalla rete, per esempio su cygwin.com c'è il compilatore g77. Oppure, da un altro sito si può scaricare il Fortran Force2.0.8.
Tutti i dettagli sono messi sul sito delle dispense.
Per poter compilare ed eseguire un programma FORTRAN, abbiamo quindi bisogno di un compilatore - sia esso gfortran o g77 o f77 o altro ancora.

Supposto di aver scritto il programma prova.f - mediante un editor di testo - per rendere eseguibile il programma, lanceremo il comando su una finestra di shell (in ambiente linux o tramite cygwin) o interattivamente con il Force2.0.8. Nel caso in cui i comandi vanno lanciati mediante una finestra di shell, si ha - per esempio:

$ g77 -o esegui prova.f

Chiamamo esegui il file eseguibile creato durante la compilazione. A questo punto possiamo eseguire il programma con il comando:

$ esegui

Oppure - a seconda della macchina -

$ ./esegui

Vediamo un esempio di programma prova.f

Primi comandi

- In FORTRAN77 tutte le istruzioni vanno scritte tra la 7-ima e la 72-sima colonna.
- le colonne da 1 a 6 hanno un significato particolare. Per ora ci interessa sapere che
  - sulla colonna 1 si mette una C o ! o un altro simbolo se poi si vuole scrivere un commento
  - sulla colonna 6 si mette un carattere qualsiasi (meglio lettere dell'alfabeto) se l'istruzione che si è scritta sulla riga precedente è troppo lunga e si deve andare a capo.

Esempio. Torniamo al programma prova.f

Istruzioni program e end

Il programma che abbiamo scritto inizia con `program prova`.
Stiamo scrivendo un programma e diamo ad esso un nome.
Il programma termina con l'istruzione `end`.
Tutti i programmi devono iniziare con l'istruzione program e finire con l'istruzione end

Operare con le variabili

Vogliamo scrivere un programma che trasforma un angolo da gradi a radianti. Abbiamo bisogno dell'angolo espresso in gradi, primi e secondi e dobbiamo restituire il valore in radianti. A tal fine abbiamo bisogno di 4 variabili, che chiamiamo `xgradi, xprimi, xsecondi, xradianti`.
Ciascuna di queste variabili occuperà un posto - o locazione - di memoria nel calcolatore. A seconda che la variabile sia dichiarata poi intera, reale o complessa, cambia il tipo di locazione di memoria che tale variabile occuperà.
Nel nostro caso, dichiariamo come intere le variabili relative all'angolo in grado e come reale la `xradianti`.
Siccome vogliamo i risultati in doppia precisione, scriveremo `real*8`.
Vediamo il programma

Osservazioni

- L'istruzione `implicit none` dice che nessuna variabile può essere dichiarata implicitamente. Quindi dobbiamo dichiarare tutte le variabili, altrimenti avremo un messaggio di errore durante la compilazione. É importante dichiarare tutte le variabili per evitare errori - per esempio se una variabile é dichiarata come `xold` e poi scriviamo `x0ld`, con l'implicit none ci viene segnalato l'errore. Senza implicit none non ci viene segnalato l'errore ma i risultati che avremo saranno inattendibili perchè non possiamo sapere quale valore sarà stato dato alla variabile `x0ld`.
- I valori delle variabili sono date mediante i comandi di `write` e di `read` - mediante terminale, da tastiera.
- Il risultato è stampato sulla finestra di shell.

- Osserviamo che le "frasi" che vogliamo mostrare sulla finestra di terminale sono scritte tra apici (usare il simbolo dell'apostrofo ').
- Vedremo che sarà molto più utile leggere i dati di input da un file esterno e scrivere i dati di output su un file.
- Per scrivere $\pi = 3.14159265\cdots$ utilizziamo la formula `2.0d0*dasin(1.0d0)` dove `dasin` è una funzione intrinseca del FORTRAN che fornisce l'arcoseno in doppia precisione.
- Tutti i numeri interi li abbiamo scritti con l'estensione .0d0 quando il risultato deve essere in doppia precisione, in modo da avere risultati più accurati.

I tipi di dati

Le variabili possono essere intere, reali, logiche (il risultato può essere solo vero o falso), complesse, di carattere (stringhe di caratteri).
Al momento ci interessa:

`integer`	variabile intera
`real`	variabile reale in precisione semplice
`real*8`	variabile reale in doppia precisione occupa esattamente 8 byte di memoria
`double precision`	variabile reale in doppia precisione accuratezza diversa a seconda della macchina usata
`logical`	variabile logica

Espressioni aritmetiche

- $**$ elevamento a potenza
- $*$ moltiplicazione
- $/$ divisione
- $+$ addizione
- $-$ sottrazione

Prima si fanno gli elevamenti a potenza, poi le moltiplicazioni e divisioni, alla fine le addizioni e sottrazioni.
Quando due operatori hanno la stessa priorità vengono eseguite le operazioni partendo da sinistra e andando verso destra.
Tuttavia, quando le espressioni sono abbastanza complicate e c'è il rischio di non capire bene quali operazioni vanno fatte prima e quali dopo, conviene mettere sempre le parentesi.