Brevi note per l'utilizzo della doppia precisione nel Fortran 90

Nei programmi che realizzano algoritmi legati al Calcolo Numerico ed ai metodi relativi, è impensabile poter lavorare con costanti e variabili reali in *singola precisione* (memorizzate di norma in 32 bit, e quindi con al più 7 cifre significative decimali). Ogni costante e variabile reale **deve** essere definita almeno in **doppia precisione**, utilizzando di norma 64 bit e potendo quindi rappresentare circa 15-16 cifre decimali significative.

Costanti reali in doppia precisione

Quando in un programma Fortran si scrive una costante reale nella forma *fixed point* oppure *floating point*, essa viene trattata come costante in *semplice precisione*. Se vogliamo che la stessa costante sia in *doppia precisione* dobbiamo scriverla sempre in forma esponenziale, utilizzando la lettera `D` (oppure `d`), anzichè la lettera `E` (oppure `e`).
Esempio:

```
Semplice precisione     Doppia precisione
-12.                    -12.D0
-12.0                   -12.0d0
47.3                    47.3D0
-12.E2                  -12.D2
56.45E0                 56.45D0
0.0                     0.0d0
```

Variabili reali in doppia precisione

Il tipo di default assegnato ad una variabile dichiarata `REAL` è la semplice precisione. Il compilatore Fortran 90 assegna un valore intero differente (detto `KIND`) alle variabili reali in semplice e in doppia precisione.
I valori interi che distinguono i due tipi di variabili non sono gli stessi per tutti i compilatori Fortran 90. Il semplice programma che segue (tratto dal testo di S.J. Chapman) permette di determinare tali due valori

```
PROGRAM kinds
IMPLICIT NONE
WRITE(*,'("KIND per singola precisione ",I2)') KIND(0.0)
WRITE(*,'("KIND per doppia precisione  ",I2)') KIND(0.0D0)
END PROGRAM kinds
```

Il compilatore **gfortran** restituisce i seguenti risultati

```
KIND per singola precisione  4
KIND per doppia precisione   8
```

A questo punto, uno dei modi possibili per dichiarare variabili in semplice e doppia precisione è mostrato nel seguente esempio:

```
PROGRAM testkind
IMPLICIT NONE
INTEGER, PARAMETER :: single = 4
INTEGER, PARAMETER :: double = 8
REAL(single) :: X
REAL(double) :: Y
X = 1./3.
WRITE(*,'(A, E22.16)') 'X = ', X
X = 1.d0/3.d0
WRITE(*,'(A, E22.16)') 'X = ', X
Y = 1./3.
WRITE(*,'(A, E22.16)') 'Y = ', Y
Y = 1.d0/3.d0
WRITE(*,'(A, E22.16)') 'Y = ', Y
END PROGRAM testkind
```

I valori che si ottengono sono

```
X = 0.3333333432674408E+00
X = 0.3333333432674408E+00
Y = 0.3333333432674408E+00
Y = 0.3333333333333333E+00
```

Come si vede, solo l'ultimo valore è un'approssimazione con 16 cifre significative esatte del numero $0.\overline{3}$.
Perchè?