INTERPOLAZIONE DI LAGRANGE

```
      program lagran
c utilizza una funzione
      parameter (n=4,np=8)
      real x(10), y(10), xx(10)
      data x/1.,2.,3.,4.,6*0./,y/1.,2.5,9.1,16.3,6*0./
      data xx/0.5,1.,1.5,2.,2.5,3.,3.5,4.,2*0./

      write(*,*) 'Dati di riferimento'
      write(*,*) '  x           y'
      do 5 i=1,n
        write(*,'(2F10.6)') x(i),y(i)
5     continue

      write(*,*) 'Dati interpolati'
      write(*,*) '  x           y-calc'
      do 20 i=1,np
        ycalc = 0.
        do 10 j=1,n
          ycalc = ycalc + y(j)*pl(j,xx(i),x,n)
10      continue
        write(*,'(2F10.6)') xx(i),ycalc
20    continue
      end

      real function pl(i,x,xn,n)
      real xn(n), num, den
      num = 1.
      den = 1.
      do 10 j=1,n
        if (i.ne.j) then
          num = num*(x    -xn(j))
          den = den*(xn(i)-xn(j))
        endif
10    continue
      pl = num/den
      return
      end
```

```
      program lagran
C variante con subroutine del programma LAGRAN
      parameter (n=4,np=8)
      real x(10), y(10), xx(10)
      data x/1.,2.,3.,4.,6*0./,y/1.,2.5,9.1,16.3,6*0./
      data xx/0.5,1.,1.5,2.,2.5,3.,3.5,4.,2*0./

      write(*,*) 'Dati di riferimento'
      write(*,*) '  x             y'
      do 5 i=1,n
        write(*,'(2F10.6)') x(i),y(i)
5     continue

      write(*,*) 'Dati interpolati'
      write(*,*) '  x          y-calc'
      do 20 i=1,np
        call lagran(xx(i),x,y,n,ycalc)
        write(*,'(2F10.6)') xx(i),ycalc
20    continue
      end

      subroutine lagran(x,xn,yn,n,ycalc)
      real xn(n), yn(n), num, den
      ycalc = 0.
      do 20 i=1,n
        num = 1.
        den = 1.
        do 10 j=1,n
          if (i.ne.j) then
            num = num*(x    -xn(j))
            den = den*(xn(i)-xn(j))
          endif
10      continue
        ycalc = ycalc + yn(i)*num/den
20    continue
      return
      end
```

```
      PROGRAM D3R1
c     Interpolazione di Lagrange con metodo avanzato
C     Programma pilota per la routine POLINT
      PARAMETER(NP=10,PI=3.1415926)
      DIMENSION XA(NP),YA(NP)
      WRITE(*,*) 'Generazione della tavola di interpolazione'
      WRITE(*,*) ' ... sin(x)    0<x<pi'
      WRITE(*,*) ' ... exp(x)    0<x<1 '
      WRITE(*,*) 'Quanti dati in queste tavole? (nota: N<10)'
      READ(*,*) N
      DO 14 NFUNC=1,2
            IF (NFUNC.EQ.1) THEN
                WRITE(*,*) 'La funzione seno da 0 a pi'
                DO 11 I=1,N
                     XA(I)=I*PI/N
                     YA(I)=SIN(XA(I))
11              CONTINUE
            ELSE IF (NFUNC.EQ.2) THEN
               WRITE(*,*) 'La funzione esponenziale da 0 a 1'
               DO 12 I=1,N
                     XA(I)=I*1.0/N
                     YA(I)=EXP(XA(I))
12             CONTINUE
            ELSE
                    STOP
            ENDIF
            WRITE(*,'(T10,A1,T20,A4,T28,A12,T46,A6)')
     *              'x','f(x)','interpolata','errore'
            DO 13 I=1,10
                    IF (NFUNC.EQ.1) THEN
                            X=(-0.05+I/10.0)*PI
                            F=SIN(X)
                    ELSE IF (NFUNC.EQ.2) THEN
                            X=(-0.05+I/10.0)
                            F=EXP(X)
                    ENDIF
                    CALL POLINT(XA,YA,N,X,Y,DY)
                    WRITE(*,'(1X,3F12.6,E15.4)') X,F,Y,DY
13          CONTINUE
            WRITE(*,*) '************************************'
            WRITE(*,*) 'Premi RETURN'
            READ(*,*)
14    CONTINUE
      END
```

```
      SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
      PARAMETER (NMAX=10)
      DIMENSION XA(N),YA(N),C(NMAX),D(NMAX)
c si cerca l'indice del nodo piu' vicino a X
c e si inizializza la tabella C e D
      NS=1
      DIF=ABS(X-XA(1))
      DO 11 I=1,N
        DIFT=ABS(X-XA(I))
        IF (DIFT.LT.DIF) THEN
          NS=I
          DIF=DIFT
        ENDIF
        C(I)=YA(I)
        D(I)=YA(I)
11    CONTINUE
c questa e' l'approssimazione iniziale a Y
      Y=YA(NS)
      NS=NS-1
      DO 13 M=1,N-1
        DO 12 I=1,N-M
          HO=XA(I)-X
          HP=XA(I+M)-X
          W=C(I+1)-D(I)
          DEN=HO-HP
          IF(DEN.EQ.0.)PAUSE 'errore'
c questo errore si verifica solo si nodi XA sono uguali
          DEN=W/DEN
          D(I)=HP*DEN
          C(I)=HO*DEN
12      CONTINUE
c si sceglie come percorrere la tabella per aggiornare
c il valore di Y (il ramo scelto e' quello piu' retto)
        IF (2*NS.LT.N-M)THEN
          DY=C(NS+1)
        ELSE
          DY=D(NS)
          NS=NS-1
        ENDIF
        Y=Y+DY
13    CONTINUE
      RETURN
      END
```