

Calcolo del fattore di convergenza

Dato uno schema iterativo si ha:

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|^p} = M$$

p è l'ordine di convergenza del metodo iterativo

M è la costante asintotica dell'errore o fattore di convergenza.

Metodo del Punto Fisso

Il metodo del punto fisso è un metodo iterativo **lineare**:



$$p = 1 \text{ e } M = |g'(\xi)|$$

dove g è la funzione di cui si vuole cercare il punto fisso ξ .

Come calcolare M nel metodo del Punto Fisso

In generale, non si conosce il punto ξ e quindi non possiamo applicare la formula precedente (il limite per $k \rightarrow \infty$) per calcolare M .

Con l'esecuzione del codice FORTRAN sul metodo del punto fisso, siamo in grado di calcolare $|x_{k+1} - x_k|$ ma non $|x_{k+1} - \xi|$.

Valutiamo lo scarto ma non l'errore che commettiamo.

Lo scarto può essere preso come una buona approssimazione dell'errore, quando ci si avvicina a convergenza:

$$|x_{k+1} - x_k| \approx |x_{k+1} - \xi|.$$

Allora,

$$\frac{|x_{k+1} - x_k|}{|x_k - x_{k-1}|} \approx M$$

Modifiche da fare al codice FORTRAN

Sostituiamo la variabile `scarto` con `scartokp1 = |xk+1 - xk|` (il significato delle due variabili è lo stesso). Aggiungiamo la variabile `scartok = |xk - xk-1|`. Ad ogni passo del metodo iterativo possiamo calcolare un'approssimazione di M mediante `scartokp1/scartok`.

Inoltre, dalla teoria, sappiamo che, per il metodo del punto fisso, $M = |g'(\xi)|$: possiamo approssimare M andando a calcolare $g'(x_k)$ ad ogni iterazione. (Quindi dobbiamo conoscere la funzione g').

Ci aspettiamo che, quando arriviamo a convergenza le due approssimazioni ottenute per M siano molto vicine tra loro.

Implementazione in FORTRAN

Alla stessa maniera con cui usiamo x_k e x_{kp1} , avremo $scartok$ e $scartokp1$, gli scarti alla iterata precedente e alla nuova iterata, rispettivamente.

Introduciamo altre due nuove variabili che rappresentano un'approssimazione di M , che chiamiamo $asint1$ e $asint2$.

$asint1 = scartokp1/scartok =$ rapporto tra due scarti successivi

$asint2 = abs(dfun(xkp1)) =$ valore assoluto della derivata prima della funzione fun in $xkp1$ - nel programma bisogna aggiungere una `function` che calcola il valore della derivata prima di fun , che chiamiamo `dfun`.

Il ciclo `do while` diventa:

```
do while ((scartokp1.ge.toll).and.(iter.le.itmax))
  iter = iter +1
  xkp1=fun(xk)
  scartokp1=abs(xkp1- xk)
  asint1= scartokp1/scartok
  asint2=abs(dfun(xkp1))
  write(*,*) iter,xk,xkp1,scartokp1,asint1, asint2
  xk=xkp1
  scartok=scartokp1
end do
```

Osservazione

Dai risultati che avremo, osserveremo delle "differenze numeriche" tra `asint1` e `asint2`.

Proviamo a cambiare il tipo di dichiarazione delle variabili reali: non più in semplice precisione `real` ma in doppia precisione `real*8`. Queste "differenze" non ci saranno più proprio perchè abbiamo una maggiore precisione nella rappresentazione dei numeri reali (e quindi di `xk`, `xkp1`, `scartok`, ...)

Esempio (con $\cosh x/2$, $x_0 = 0$, $\text{toll}=1.e-6$)

Variabili reali di tipo real

| iter | xk | xkp1 | scarto | asint1 | asint2 |
|------|------------|------------|----------------|-------------|-------------|
| 9 | 1.17872989 | 1.17876124 | 3.13520432E-05 | 0.311981022 | 0.312050164 |
| 10 | 1.17876124 | 1.17877102 | 9.77516174E-06 | 0.311787069 | 0.312053055 |
| 11 | 1.17877102 | 1.17877412 | 3.09944153E-06 | 0.317073166 | 0.312053978 |
| 12 | 1.17877412 | 1.17877507 | 9.53674316E-07 | 0.307692319 | 0.312054247 |

Variabili reali di tipo real*8

| iter | xk | xkp1 | scarto | asint1 | asint2 |
|------|------------|------------|----------------|-------------|-------------|
| 9 | 1.17872992 | 1.1787613 | 3.13751481E-05 | 0.312026124 | 0.312050186 |
| 10 | 1.1787613 | 1.17877109 | 9.79047574E-06 | 0.312045563 | 0.312053071 |
| 11 | 1.17877109 | 1.17877414 | 3.05513389E-06 | 0.312051628 | 0.312053971 |
| 12 | 1.17877414 | 1.17877509 | 9.53365289E-07 | 0.312053521 | 0.312054252 |

Metodo di Newton-Raphson

Il metodo di Newton-Raphson è uno schema iterativo per la ricerca degli zeri di una funzione $f(x)$.

Rispetto al metodo del punto fisso che abbiamo visto prima, il metodo di Newton-Raphson non cerca un punto fisso della funzione f ma uno zero della f , vale a dire un punto ξ tale che $f(\xi) = 0$.

Algoritmo

Data un'approssimazione iniziale x_0 , si ha:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Si va avanti nel calcolo delle approssimazioni x_{k+1} fino a che lo scarto $|x_{k+1} - x_k|$ diventa minore di una certa tolleranza `toll` (si segue lo stesso ragionamento fatto per il metodo del punto fisso).

Ordine del metodo

Dalla teoria, sappiamo che generalmente il metodo di Newton-Raphson è del **secondo ordine**:

$$\Downarrow$$
$$p = 2 \text{ e } M = \frac{|f''(\xi)|}{|2f'(\xi)|}$$

Stima del fattore di convergenza

Utilizzando gli scarti `scartokp1` e `scartok`, possiamo stimare un'approssimazione di M , mediante la formula

$$\frac{|x_{k+1} - x_k|}{|x_k - x_{k-1}|^2} \approx M$$

Nel codice

```
asint = scartokp1/scartok**2
```

(La procedura è del tutto equivalente a quanto fatto nel metodo del punto fisso).

Volendo si può calcolare un'approssimazione di M anche applicando direttamente la formula con le derivate della funzione (**facoltativo**).

Funzioni da applicare al codice

Per testare il codice FORTRAN, possiamo considerare le funzioni $g(x)$ di cui abbiamo cercato il punto fisso nel programma del metodo del punto fisso e considerare le nuove funzioni $f(x) = g(x) - x$.

Uno zero della f è un punto fisso della g e viceversa.

Difatti, se ξ è punto fisso della funzione g , allora

$$g(\xi) = \xi \iff g(\xi) - \xi = 0 \iff f(\xi) = 0$$

Esempi di funzioni

| $g(x)$ | $f(x)$ |
|------------------------|----------------------------|
| $\cos(x)$ | $\cos(x) - x$ |
| $\sqrt{e^{-x^2}}$ | $\sqrt{e^{-x^2}} - x$ |
| $e^{-x/4}$ | $e^{-x/4} - x$ |
| $\cosh(x/2)$ | $\cosh(x/2) - x$ |
| $x^2 - 5x + 9$ | $(x - 3)^2$ |
| $1 + \mathbf{atan}(x)$ | $1 + \mathbf{atan}(x) - x$ |