Università degli Studi di Padova

Dipartimento di Ingegneria Industriale

Dipartimento Matematica

Dipartimento di Matematica "Tullio Levi-Civita"

Laboratorio di Calcolo Numerico LAB 5

Metodo di Newton e stime di convergenza

Docenti: E. Bachini, L. Bruni

Email: elena.bachini@unipd.it Email: bruni@math.unipd.it

3 aprile 2024

Metodo di Newton

METODI PER LA RISOLUZIONE DI EQ. NON LINEARI

Anche in questo laboratorio ci occuperemo di metodi che ci permettano di individuare gli zeri di una data funzione

$$f : \mathbb{R} \to \mathbb{R}$$

(assumendo opportune ipotesi di regolarità della funzione)
ossia i valori dell'incognita ξ per cui $f(\xi) = 0$.

Metodo di Newton

METODI PER LA RISOLUZIONE DI EQ. NON LINEARI

- Lab4. **Metodo di Bisezione**: usa solamente il valore della funzione in una serie di punti (gli estremi degli intervalli considerati)
- Lab5. **Metodo di Newton**: usa non solo il valore della *funzione*, ma anche quello della sua 1^a *derivata* (che dovrà perciò esistere nell'intervallo considerato)

Metodo di Newton

Data $f : \mathbb{R} \to \mathbb{R}$ continua e **derivabile** (derivata f' continua!!), trovare ξ tale che $f(\xi) = 0$
Vogliamo costruire una successione

$$x_{k+1} = x_k + h_k$$

Pensiamo che $x_{k+1} = \xi$ e quindi $f(x_{k+1}) = 0$. Essendo f' continua, usando le serie di Taylor:

$$0 = f(x_{k+1}) = f(x_k + h_k) = f(x_k) + h_k \cdot f'(x_k) + o(2)$$

Da cui trascurando termini di ordine superiore al primo
$0 = f(x_k) + h_k \cdot f'(x_k)$ e quindi

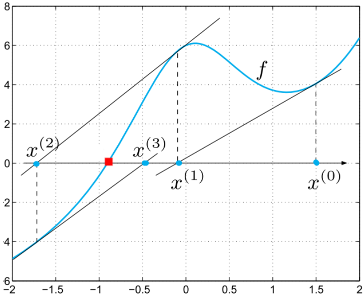
$$h_k = -\frac{f(x_k)}{f'(x_k)}$$

Metodo di Newton

Quindi, partendo da un punto arbitrario x_0 la k-esima iterazione

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \qquad k = 0, 1, \ldots$$

Linearizziamo iterativamente la funzione sostituendo f ad ogni iterazione con la sua tangente nel punto $(x_k, f(x_k))$ e identificando con x_{k+1} lo zero di questa retta tangente.



Esercizio 1 (con consegna)

Gli obiettivi di questo esercizio sono:

- la creazione di una funzione `newton.m` che implementi l'algoritmo di Newton per trovare lo zero di una funzione continua
- la creazione di uno script `CN_lab5_es1.m` in cui
  - si definiscono i dati di input della funzione `newton.m`
  - si esegue la funzione `newton.m`
  - si visualizzano i risultati della stessa

Es. 1 - Metodo di Newton

Parto da x_0

1. calcolo $f(x_k)$ e $f'(x_k)$. Se la derivata è uguale a zero: STOP
2. calcolo $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
3. se $|x_{k+1} - x_k| < toll$ STOP
4. altrimenti ($|x_{k+1} - x_k| > toll$) ripeti punto 1

Es. 1 - Pseudocodice Metodo di Newton

Di seguito lo pseudo-codice della funzione che implementa il Metodo di Newton

```
[x,fx,n]=newton(f,f1,x0,toll,nmax)

x=[]; fx=[];
n=1; diff=2*toll;
while (abs(diff) >= toll) & (n < nmax)  do
   if f1(x(n)) == 0 then
     error
   else
      diff=-f(x(n))/f1(x(n));
      x(n+1)=x(n)+diff;
      update x; fx;
      n=n+1;
    end if
end while
```

Es. 1 - Intestazione function `newton.m`

Si implementi in Matlab la funzione `newton.m` che implementi l'algoritmo di Newton ed abbia la seguente intestazione:

```
function [xv, fxv, n] = newton (f, f1, x0, toll, nmax)
%NEWTON Metodo di Newton
%
% [xv, fxv, n] = newton (f, f1, x0, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione
%   f1:     derivata prima
%   x0:     valore iniziale
%   toll:   tolleranza richiesta per il valore assoluto
%           della differenza di due iterate successive
%   nmax:   massimo numero di iterazioni permesse
%
% Dati di uscita:
%   xv:     vettore contenente le ascisse delle iterazioni
%   fxv:    vettore contenente i corrispondenti residui (f(xv))
%   n:      numero di iterazioni
```

Es 1 Script CN_lab5_es1.m

Si scriva lo script CN_lab5_es1.m seguendo il template qui riportato

```
%==========================================================
% Script per il Metodo di Newton
% Necessita della Function newton.m
%==========================================================
clear all
clc
disp('METODO DI NEWTON');

% Ingresso dati
% funzione e ua derivata:
f =
f1 =

% punto iniziale x0
x0 = ...

%-------------------
%tolleranza
toll = 1e-8;
%numero massimo iterazioni
nmax = 100;
%-------------------
```

Es 1 - Script CN_lab5_es1.m

```
% visualizzazione di controllo dati inseriti
fprintf('----------DATI----------\n');
disp('f =  ');
disp(f);
disp('derivata di f =  ');
disp(f1);
fprintf('x0 = %f \n', x0);
fprintf('toll = %f \n', toll);
fprintf('n. max iterazioni = %f \n', nmax);
```

```
% Parte esecutiva
% chiamata function newton
[xv, fxv, n] = newton(f, f1, x0, toll, nmax);
% dai un warning in caso di numero massimo di iterazioni nmax raggiunto
...
% altrimenti scrivi che la convergenza é stata raggiunta
...
```

Es 1 - Script CN_lab5_es1.m

```
% Risultati
fprintf('————RISULTATI————\n');
% Visualizza l'ultima iterata
...
% Visualizza l'ultimo valore della funzione
...
% Visualizza il numero di iterazioni
...


% Grafico dei valori assoluti dei residui (valori della funzione vs
% numero dell'iterazione (in scala logaritmica)
...

% Scrivi una tabella con i valori delle ascisse e delle funzioni
% calcolate alle varie iterazioni
fprintf('      xn,              fxn \n')
fprintf (...)
```

Consegna - Lab5

Dopo aver testato l'algoritmo con semplici funzioni note:

- Si usino le 4 funzioni proposte nell'esercizio 1 del Lab4 come test (impostando `toll` = 10^-8 e `nmax` = 100).
- Si adatti opportunamente lo script perchè corra sia l'algoritmo di Newton che quello di Bisezione (Lab4) e crei un grafico sovrapposto di convergenza (si usi `hold on`).
- Si salvino le immagini ottenute (NON in formato .fig)
- Si crei un file .zip con i codici creati (`CN_lab5_es1.m` e `newton.m`) e con i grafici di convergenza ottenuti e opportunamente modificati. Si rinomini lo zip (**cognome_nome_lab5.zip**) e si carichi su moodle

Spunti di riflessione

Nota.

Che ordine di convergenza ottenete? Notate la differenza tra l'algoritmo di Bisezione e di Newton.

Nota.

Considerate la funzione $f = (x-1)^2$ *e* $x_0 = 1.5$. *Che ordine di convergenza ottenete usando Newton? Perchè? L'algoritmo di bisezione non funziona. Perchè?*

Nota.

Non sempre è possibile calcolare la derivata della funzione e si tende ad sostiture la derivata prima con un rapporto incrementale tra le funzioni e le ascisse.

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

ottenendo il ***Metodo della secante***
Come si dovrebbe modificare l'algoritmo per ottenere il metodo della secante?

Stima dell'ordine di convergenza

Se $\{x_n\} \to x$ al tendere di $n \to \infty$, l'ordine di convergenza p può essere stimato con l'approssimazione:

$$p_n = \frac{\log \left| \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}} \right|}{\log \left| \frac{x_{n-1} - x_{n-2}}{x_{n-2} - x_{n-3}} \right|}, qquad n \geq 4$$

Si può provare che $p_n \to p$.

- Data $\{x_n\}$, possiamo calcolare p_n con un ciclo `for`, oppure in modo più efficiente utilizzando le opportune operazioni su vettori.
- La successione che otteniamo dai nostri metodi iterativi è finita. Quindi l'algoritmo che dovrà calcolare la successione delle stime p_n, dovrà testare con una `if` che la successione delle iterate abbia almeno 4 componenti.

Function `stimap` che implementa la stima dell'ordine

```
function pv = stimap (xv, nt)
%STIMAP  Stima dell'ordine di convergenza di una successione
%
% Uso:
%   pv = stimap (xv, nt)
%
% Dati di ingresso:
%   xv:     vettore colonna contenente le iterate della successione.
%   nt:     numero dei termini della successione da considerare per la stima.
%
% Dati di uscita:
%   pv:      vettore contenente le stime dell'ordine calcolate.
%           Le prime tre componenti del vettore sono sempre nulle.
```

ATTENZIONE: La prima stima calcolata è p_4 che utilizza le iterate x_1, x_2, x_3 e x_4. Pertanto le prime tre componenti del vettore `p` saranno sempre nulle.

Esercizio 2 (senza consegna)

Gli obiettivi di questo esercizio sono:

- la creazione di un algoritmo e poi una function `stimap` che permetta di costruire la stime dell'ordine di convergenza a partire da una successione data, utilizzando la formula precedente (slide 14)
- utilizzare tale funzione per calcolare le stime per la successione ottenute con i metodi di Bisezione e di Newton
- visualizzare un grafico con le coppie di valori $(n, |p_n|)$, dei valori assoluti delle stime al crescere di n