Esercitazione 5

Istruzioni trattate: `xgrid`, `legend`, `plot2d`, `and`.

Nella prima parte di questa esercitazione vedremo una realizzazione di un *metodo ad un punto* e la utilizzeremo per approssimare il *punto unito* della funzione $h_3 : \mathbb{R} \to \mathbb{R}$ definita da:

$$h_3(x) = \frac{x + e^{-x}}{2}$$

Nella seconda parte utilizzeremo la realizzazione per approssimare, con il *metodo di Newton*, lo *zero* della funzione $F : (0, +\infty) \to \mathbb{R}$ definita da:

$$F(x) = x - \log x$$

che coincide con il punto unito di h_3.

Prima parte

La definizione che segue è una realizzazione del *metodo ad un punto* definito dalla funzione h.

```
function [x, iter, StimaErr] = MetodoUnPunto(h,x0,tol,Nmax)
  //
  // x: colonna di numeri reali. Le componenti di questo vettore sono gli
  //    elementi calcolati della successione generata dal metodo definito
  //    da h a partire da x0.
  // iter: numero intero. E' il numero di iterazioni effettuate.
  // StimaErr: colonna di numeri reali non negativi. StimaErr(k) è la
  //           stima dell'errore assoluto commesso utilizzando x(k) per
  //           approssimare un punto unito di h.
  //
  // tol: numero reale positivo per il criterio d'arresto.
  // Nmax: numero intero positivo. E' il massimo numero di iterazioni
  //       consentito.
  //
  x = x0;
  iter = 0;
  StimaErr = abs(h(x($)) - x($));
  while StimaErr($) > tol & iter < Nmax,
        x($+1) = h(x($));
        StimaErr($+1) = abs(h(x($)) - x($));
        iter = iter + 1;
  end;
endfunction
```

In questa realizzazione si è scelto di proseguire la costruzione della successione *finché* (`while`) la stima dell'errore assoluto è maggiore della quantità `tol` assegnata dall'utilizzatore (criterio di arresto di tipo assoluto) *e* il numero di iterazioni effettuate è inferiore al numero massimo imposto dall'utilizzatore `Nmax`.

Definiamo adesso la funzione `h3`, realizzazione di h_3:

```
function y = h3(x)
  //
  // x,y: matrici ad elementi reali di uguale dimensione;
  //
  y = (x + exp(-x))/2;
endfunction
```

Prima di utilizzare il metodo ad un punto definito dalla funzione h_3 ci domandiamo se h_3 ha punti uniti e, in caso affermativo e per ciascun punto unito, se il metodo sia utilizzabile per l'approssimazione. Sappiamo già come sia possibile scoprire, *analiticamente*, che h_3 ha *un solo punto unito*, α, separato dall'intervallo $[\frac{1}{2}, 1]$, che il metodo *è utilizzabile* ed ha *ordine di convergenza uno ad* α e che *per ogni punto iniziale* $x_0 \in [\frac{1}{2}, 1]$ *si ottiene una successione convergente ad* α *e monotona*. Vediamo che informazioni si possono ottenere, invece, *graficamente*.

Riportiamo su uno stesso piano cartesiano i grafici, sull'intervallo $[0, 1]$, delle funzioni $h_3(x)$ e x:[1]

```
-->x = linspace(0,1,200)';

-->plot2d(x,[h3(x),x]);

-->xgrid(3); xlabel('x'); legend('h3(x)','x','in_upper_left');
```

Si ottiene il disegno riportato in Figura 1. L'argomento 3 nel comando `xgrid` cambia *il colore* con cui viene disegnata la griglia da nero (colore predefinito) a verde. L'opzione `'in_upper_left'` presente nel comando `legend` fa posizionare la legenda nell'angolo *in alto a sinistra* della figura.

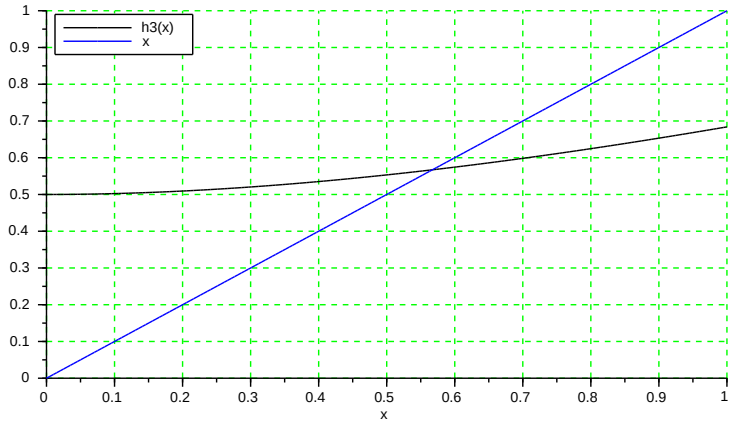

Figura 1: Studio grafico per il metodo definito da h_3.

Dal disegno si deduce che, *nell'intervallo* $[0, 1]$: (a) i grafici di $h_3(x)$ ed x si intersecano in un solo punto; (b) la retta tangente al grafico di h_3 nel punto di intersezione ha pendenza positiva e minore di uno; (c) in tutti i punti del grafico di h_3 con ascissa maggiore o uguale a 0.2, la retta tangente ha pendenza positiva e minore di uno. Ne segue che: (a) nell'intervallo $[0, 1]$ la funzione h_3 ha un solo punto unito, α; (b) $0 < |h_3'(\alpha)| < 1$ e quindi il metodo definito da h_3 è utilizzabile per l'approssimazione ed ha ordine di convergenza ad α uno; (c) per ogni $x \in [0.2\,; 1]$ si ha: $0 < h_3'(x) < 1$ e quindi *per ogni* $x_0 \in [0.2\,; 1]$ la successione generata dal metodo risulta convergente ad α e monotona.

Si osservi che dal disegno si possono dedurre informazioni relative al *solo* intervallo $[0, 1]$. Il particolare, il disegno *non fornisce informazioni sull'esistenza di altri punti uniti al di fuori di tale intervallo*. La scelta dell'intervallo da considerare per lo studio è *responsabilità dell'utilizzatore*.

Con le informazioni ottenute, analiticamente o graficamente, possiamo utilizzare la procedura `MetotoUnPunto` per ottenere un'approssimazione del punto unito α:

```
-->x0 = 1;

-->tol = 1d-9;

-->Nmax = 15;

-->[z3,iter3, StimaErr3] = MetodoUnPunto(h3,x0,tol,Nmax);
```

[1] Più correttamente: *un'approssimazione* dei grafici di $h_3(x)$ e x.

```
-->printf('\n* Metodo h3 (tol = %3.2e , Nmax = %d) *\n\n' + ...
          '  z3 = %15.14e , iter3: %d , StimaErr3 = %3.2e\n\n', + ...
          tol,Nmax,z3($),iter3,StimaErr3($));

-->plot2d(z3($),h3(z3($)),style = -5);

-->legend('h3(x)','x','(z3 , h3(z3))','in_upper_left');
```

Si ottiene, nella console:

```
* Metodo h3 (tol = 1.00e-09 , Nmax = 15) *

  z3 = 5.67143290702951e-01 , iter3: 14 , StimaErr3 = 2.30e-10
```

e nella finestra grafica il disegno riportato in Figura 2. Il valore negativo per l'opzione `style` del comando `plot2d` fa disegnare *un simbolo* con centro nelle coordinate del punto richiesto (in questo caso il simbolo $\diamond$ con centro in $(\texttt{z3}, \texttt{h3}(\texttt{z3}))$.

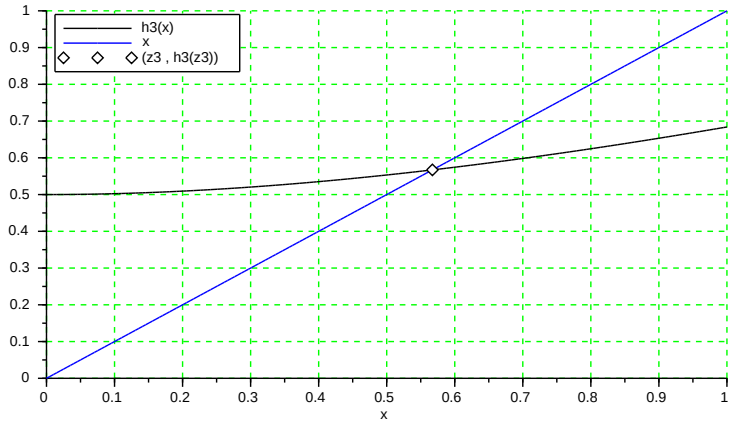

Figura 2: L'approssimazione ottenuta con il metodo definito da h_3.

Seconda parte

Il *metodo di Newton* applicato alla funzione F è il metodo ad un punto definito da:

$$h_N(x) = x - \frac{F(x)}{F'(x)}$$

Definiamo le funzioni `F` e `d1F`, realizzazioni, rispettivamente, delle funzioni F ed F' facendo attenzione al loro insieme di definizione ed all'uso dell'operatore `./` nella seconda definizione:[2]

```
function y = F(x)
  //
  // x: matrice ad elementi reali positivi;
  // y: matrice ad elementi reali di dimensione uguale a quella di x;
  //
  if ~and(x > 0) then error('F: argomento non positivo'); end;
  y = x + log(x);
endfunction
//
function y = d1F(x)
  //
```

[2] L'operatore ~ é la realizzazione in *Scilab* dell'operatore logico di negazione, *not*.

```
  // x: matrice ad elementi reali positivi;
  // y: matrice ad elementi reali di dimensione uguale a quella di x;
  //
  if ~and(x > 0) then error('d1F: argomento non positivo'); end;
  y = 1 + 1 ./ x;
endfunction
```

Per entrambe le funzioni si è scelto di *verificare* che l'argomento `x` sia una matrice ad elementi reali positivi. Si osservi che se `x` è una matrice ad elementi reali, il comando `x > 0` restituisce *una matrice* della stessa dimensione di `x` di elemento i, j definito da `x(i,j) > 0`. Ad esempio:

```
-->A = [-1,2;3,4]; A > 0
 ans  =

  F T
  T T
```

Per decidere se *tutti* gli elementi di `x` sono numeri reali positivi si verifica se *tutti* gli elementi della matrice `x > 0` hanno valore `T`. Questo si ottiene utilizzando la *funzione predefinita* `and`.

- `and`

  Questa *funzione predefinita* restituisce l'*and* di tutte le componenti di un'assegnata matrice ad elementi in $\{\,$`T`$,$`F`$\,\}$.

Definiamo adesso la funzione `h_N`, realizzazione della funzione $h_N : (0, +\infty) \to \mathbb{R}$ che definisce il *metodo di Newton*:

```
function y = h_N(x)
  //
  // x: matrice ad elementi reali positivi;
  // y: matrice ad elementi reali di dimensione uguale a quella di x;
  //
  // La funzione h_N definisce il metodo di Newton applicato ad F.
  //
  y = x - F(x) ./ d1F(x);
endfunction
```

Prima di utilizzare il metodo di Newton ci domandiamo se F ha zeri e, in caso affermativo e per ciascuno zero, se il metodo sia utilizzabile per l'approssimazione. Sappiamo già come sia possibile scoprire, *analiticamente*, che F ha *un solo zero*, α, separato dall'intervallo $[\frac{1}{2}, 1]$, che il metodo *è utilizzabile ed ha ordine di convergenza ad* α *due* e che *utilizzando come punto iniziale* $x_0 = \frac{1}{2}$ *si ottiene una successione convergente ad* α *e monotona crescente*. Vediamo che informazioni si possono ottenere, invece, *graficamente*.

Riportiamo su uno stesso piano cartesiano i grafici delle funzioni $F(x)$ e $F'(x)$ sull'intervallo $[0.1\,;\,1]$:[3]

```
-->x = linspace(0.1,1,200)';

-->clf(); plot2d(x,[F(x), d1F(x)]);

-->xgrid(3); xlabel('x'); legend('F(x)','F''(x)');
```

Si ottiene il disegno riportato in Figura 3.

Dal disegno si deduce che, *nell'intervallo* $[0.1\,;\,1]$: (a) il grafico di F interseca l'asse delle ascisse in un solo punto; (b) la retta tangente al grafico di F nel punto di intersezione ha pendenza non zero; (c) per ogni x si ha $F'(x) > 0$ e la retta tangente al grafico di F' ha pendenza negativa. Ne segue che: (a) nell'intervallo $[0.1\,;\,1]$ la funzione F ha un solo zero, α; (b) il metodo di Newton è utilizzabile per l'approssimazione ed ha ordine di convergenza ad α almeno due; (c) *per ogni* $x_0 \in [0.1\,;\,\alpha)$ la successione generata dal metodo risulta convergente ad α e monotona crescente.

[3] Più correttamente, come già osservato: *un'approssimazione* dei grafici di $F(x)$ e $F'(x)$.

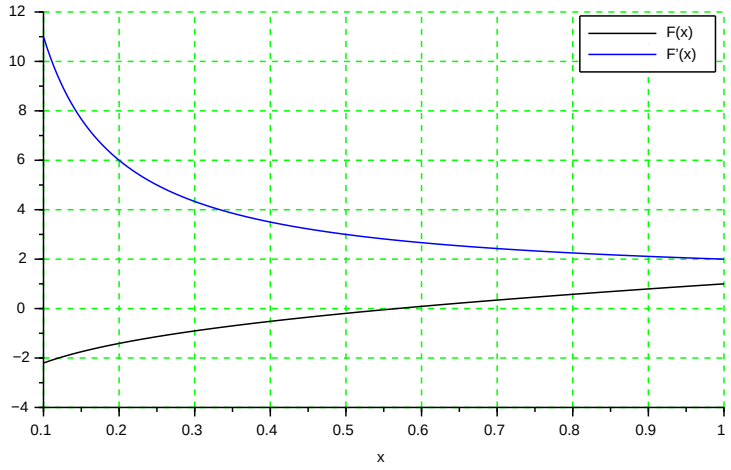

Figura 3: Studio grafico per il metodo di Newton.

Come già osservato, dal disegno si possono dedurre informazioni relative al *solo* intervallo [0.1 ; 1]. Il particolare, il disegno *non fornisce informazioni sull'esistenza di altri zeri al di fuori di tale intervallo.* La scelta dell'intervallo da considerare per lo studio è *responsabilità dell'utilizzatore.*

Con le informazioni ottenute, analiticamente o graficamente, possiamo utilizzare la procedura `MetotoUnPunto` per ottenere, con il *metodo di Newton*, un'approssimazione dello zero α:

```
-->x0 = 0.1;

-->tol = 1d-9;

-->Nmax = 30;

-->[zN,iterN,StimaErrN] = MetodoUnPunto(h_N,x0,tol,Nmax);

-->printf('\n* Metodo di Newton (tol = %3.2e , Nmax = %d) *\n\n' + ...
          '  zN = %15.14e , iterN: %d , StimaErrN = %3.2e\n\n', ...
          tol,Nmax,zN($),iterN,StimaErrN($));

-->plot2d(zN($),F(zN($)),style = -5);

-->legend('F(x)','F''(x)','(zN , F(zN))');
```

Si ottiene, nella console:

```
* Metodo di Newton (tol = 1.00e-09 , Nmax = 30) *

  zN = 5.67143290406526e-01 , iterN: 5 , StimaErrN = 3.26e-12
```

e nella finestra grafica il disegno riportato in Figura 4.

Si osservi che `zN` $\neq$ `z3`, che `StimaErrN` è molto minore di `StimaErr3` e che il metodo di Newton ha eseguito 5 iterazioni mentre il metodo definito da h_3, con lo stesso valore di `tol`, ne ha eseguite 14.

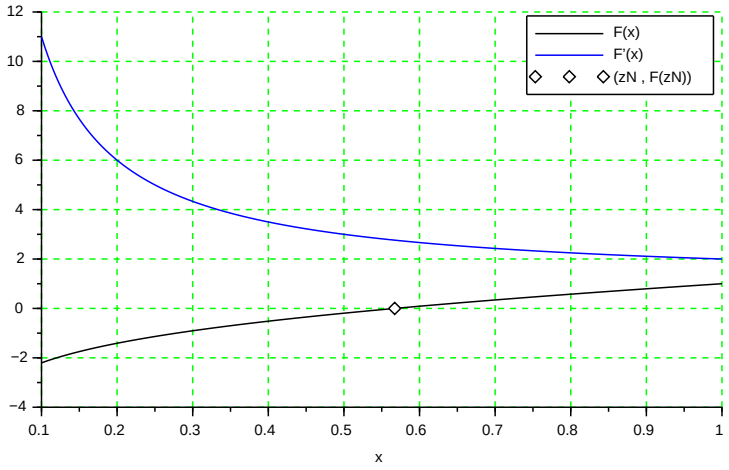

Figura 4: L'approssimazione ottenuta con il metodo di Newton.

Esercizi

1. Tenuto conto che:
$$\frac{1-\frac{1}{\sqrt{e}}}{2} \leqslant h'(x) \leqslant \frac{1-\frac{1}{e}}{2}$$
e che (dalla Figura 1):
$$0.4 \leqslant |x_0 - \alpha| \leqslant 0.5$$
verificare che il numero di iterazioni per ottenere $|x_k - \alpha| < 10^{-9}$ è compreso tra 13 e 18. Ottenere poi l'analogo intervallo relativo al numero di iterazioni per ottenere $|x_k - \alpha| < 10^{-12}$. Infine, utilizzare la realizzazione del metodo iterativo definito da h_3 con `tol` $= 10^{-12}$ e confrontare il numero di iterazioni eseguite dalla procedura con l'intervallo determinato.

2. Verificare che la parte di successione costruita utilizzando la realizzazione del metodo iterativo definito da h_3 è monotona decrescente.

3. Spiegare perché nella definizione della funzione `h_newt` *non* si effettuano le verifiche sull'argomento analoghe a quelle presenti nelle funzioni `F` e `d1F`.

4. La funzione h_N che definisce il metodo di Newton applicato ad F si può riscrivere come segue:
$$h_N(x) = \frac{x\,(1 - \log x)}{x + 1}$$
Riscrivere la definizione della funzione `h_N` utilizzando l'espressione adesso riportata. La definizione *deve* includere la verifica che la matrice argomento abbia tutte le componenti nell'opportuno insieme. Infine, utilizzare la nuova definizione per ottenere un'approssimazione dello zero α di F.

5. Riscrivere la definizione di `F` *eliminando* la verifica sull'argomento. Verificare che *Scilab* segnala un errore in risposta alla richiesta di calcolare `F(0)`, ma *non* segnala alcun errore in risposta alla richiesta di calcolare `F(-3)`.

   In *Scilab* la *funzione predefinita* `log` corrisponde alla funzione *logaritmo complesso*, definita *per ogni numero complesso* $z \neq 0$ da:
$$\log(z) = \log(|z|) + i\ \arg(z)$$

dove $\arg(z)$ è l'*argomento principale* di z (l'unico argomento di z in $(-\pi, \pi]$).[4] La verifica sull'argomento nelle funzioni `F` e `d1F` ha lo scopo di evitare fraintendimenti: se l'argomento ha qualche componente negativa, la funzione *deve* segnalarlo all'utilizzatore.

Particolare cautela occorre, in *Scilab*, quando si opera su numeri complessi con i comandi di stampa e disegno. Ad esempio, eseguire i comandi:

```
-->disp(F(-3), 'F(-3) =','Il valore di F in -3 è un numero complesso:');

-->printf('\n Il valore di F in -3 è un numero complesso:\n\n' + ...
          ' F(-3) =\n\n  %8.7f', F(-3));

-->clf(); x = linspace(-1,1,100); xN = x(1:50); xP = x(51:100);

-->plot(xN,log(xN),xP,log(xP)); xgrid(); xlabel('x'); ylabel('log(x)');
```

6. Il comando `plot2d` consente all'utilizzatore di scegliere il *tipo di scala* da usare su ciascuno degli assi cartesiani tra *lineare* e *logaritmica*, inserendo come *prima* opzione nel comando una tra le stringhe: `'nn'`, `'nl'`, `'ln'`, `'ll'`. Ciascuno dei caratteri specifica il tipo di scala da usare sul corrispondente asse (codifica: `n` significa scala lineare – *normale* –, `l` significa scala logaritmica). Eseguire i comandi:

```
-->clf(); plot2d('nl',[0:iter3]',StimaErr3, style = 3);

-->plot2d('nl',[0:iterN]',StimaErrN, style = 5);

-->xlabel('k'); legend('StimaErr3','StimaErrN');
```

   Spiegare come mai la curva corrispondente a `StimaErr3` è molto simile ad un tratto di retta e constatare, dalla figura ottenuta, che la successione generata dal metodo di Newton converge *più rapidamente* di quella generata dal metodo definito da h_3.

7. Si consideri la funzione:
$$h_2(x) = e^{-x}$$
   Utilizzare la procedura `MetodoUnPunto` per determinare, con il metodo definito da h_2, un'approssimazione `z2` del punto unito di h_2 con errore assoluto minore di 10^{-7}. Confrontare poi *la rapidità di convergenza* delle successioni costruite dai metodi definiti da h_2 e h_3 riportando su uno stesso disegno i grafici, con asse delle ordinate logaritmico, delle rispettive stime dell'errore assoluto.

[4] Per approfondire, vedere, ad esempio: `https://en.wikipedia.org/wiki/Logarithm#Complex_logarithm`.