Università
degli Studi
di Padova

A.A. 2019 - 2020

Dipartimento di Ingegneria Industriale DII

Laurea in Ingegneria Aerospaziale - Canale B

Corso di CALCOLO NUMERICO

Prima prova di accertamento MATLAB

RELAZIONE FINALE

Studente:
Oliveri Marco
n. matr. 1219903

27 aprile 2020

Premessa

Nel corso delle prime 6 (sex) lezioni telematiche di laboratorio di Calcolo Numerico svolte dal docente Prof. Bergamaschi Luca, si sono studiate le basi del software di calcolo numerico e di analisi statistica MATLAB creato dalla MathWorks.
In queste lezioni laboratoriali si sono studiati, sulla base teorica precedentemente appresa tramite le lezioni telematiche di teoria, i metodi per la risoluzione delle equazioni non lineari del tipo $f(x) = 0$.
Al termine di questa prima parte didattica, si è svolta la seguente prima prova di accertamento valida per il corso di Calcolo Numerico per la Laurea in Ingegneria Aerospaziale (Canale B).
Con la presente Relazione Finale, di seguito, verranno analizzati gli esercizi proposti dal docente del corso nella prova svolta a distanza.

ESERCIZIO 1

1. Introduzione

In questo esercizio è stato implementato uno dei metodi iterativi studiati per la risoluzione di equazioni non lineari. Nello specifico il metodo iterativo analizzato in questo esercizio, attraverso il software MATLAB, è il *Metodo di Steffensen*.

1.1 Testo esercizio

Si scriva una function MATLAB che implementi il *Metodo di Steffensen*. Si risolva l'esercizio partendo dalla function `newton.m` e rinominandola `steffensen.m`, modificare opportunamente il *Metodo di Newton* per ottenere quello di *Steffensen*.

1.2 Scopo

Utilizzare la function `steffensen.m` implementata per la soluzione iterativa di una generica equazione non lineare $f(x) = 0$ con il *Metodo di Steffensen*.

1.3 Cenni teorici

Per risolvere $f(x) = 0$, fissato un punto iniziale x_0, si considera l'iterazione del *Metodo di Steffensen*:

$$x_{k+1} = x_k - \frac{f(x_k)^2}{f(x_k + f(x_k)) - f(x_k)}$$

Ordine di convergenza: $p = 2$
Implementazione: essa è singola (passaggio da x_k a x_{k+1}) e la si può schematizzare nel seguente modo:

$$fa = f(x_k)$$

$$x_{k+1} = x_k - \frac{f_a^2}{f(x_k + f_a) - f_a}$$

2. Svolgimento

Sulla base dell'implementazione del *Metodo di Newton* sul software MATLAB della function `newton.m`, si è eseguita la nuova implementazione richiesta della function `steffensen.m`, la quale sfrutta il *Metodo di Steffensen* di iterazione. L'esercizio è stato svolto tramite semplici cambiamenti sull'editor di testo del programma sulla base teorica riassunta alla sottosezione 1.3 (pagina 2). Al termine dell'implementazione è stato editato uno script che, richiamando la function `steffensen.m`, ha dato la possibilità di controllare la function stessa fornendo una visualizzazione dei risultati e dei grafici.

3. Dati, risultati e grafici

La nuova function implementata è la seguente:

```
[iterS,xnew,scartiS] = steffensen(x0,tolS,itmaxS,fun)
```
(1)

3.1 Dati e risultati

Sulla base della function (1), tramite lo script di controllo, si è definita come esempio la funzione di input:

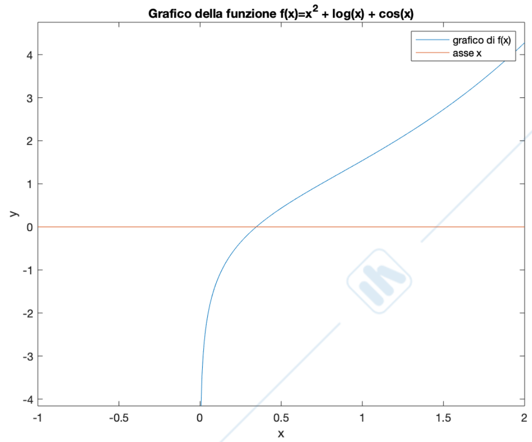
$$f(x) = x^2 + log(x) + cos(x)$$

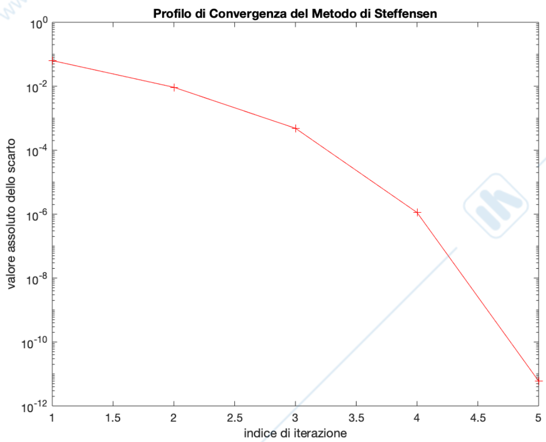
con i seguenti parametri: `[x0 = 0.4; tolS = 1e-10; itmaxS = 100]`.
I risultati ottenuti all'ultima iterazione sono i seguenti:

`iter(S)`	`xk(S)`	`|sk|(S)`
5	0.346266409996	6.00658e-12

3.2 Grafici

Dalla figura a sinistra si può osservare il grafico della $f(x)$ presa come esempio nell'intervallo I = [-1,2]. Dal grafico del profilo di convergenza (a destra), invece, si può osservare in che maniera il *Metodo di Steffensen* converge ed in questo caso in 5 iterazione.





4. Conclusioni

A differenza degli altri metodi iterativi, così come il *Metodo di Newton*, il *Metodo di Steffensen* implementato conserva l'ordine $p = 2$ e non richiede però l'utilizzo della derivata della funzione[1]. Inoltre, attraverso la teoria sulle equazioni non lineari, si può dire che il metodo analizzato è localmente convergente con ordine $p = 2$.

[1] Solitamente in corso di programmazione su MATLAB la funzione derivata è chiamata `dfun`.

ESERCIZIO 2

1. Introduzione

Nell'esercizio proposto è stato implementato, tramite il software MATLAB, il *Metodo di Bisezione*. Esso è uno fra i metodi iterativi studiati per la risoluzione di equazioni non lineari.

1.1 Testo esercizio

Si implementi il metodo iterativo della bisezione. Si usi un test di arresto sul residuo pesato (rp) e sul numero massimo di iterazioni. La sintassi della function sarà la seguente:

```
[iter,cnew,vres] = bisezione(a,b,tol,itmax,fun,dfun)
```
(2)

Si testi la correttezza della function bisezione sull'equazione $x^2 - 2 = 0$ con parametri: `[a = 1; b=2; tol = 1e-12; itmax = 50]`.

1.2 Scopo

Utilizzare la function `bisezione.m` implementata per la soluzione iterativa di una generica equazione non lineare $f(x) = 0$ con il *Metodo di Bisezione*.

1.3 Cenni teorici

Per risolvere $f(x) = 0$, fissati i parametri `a,b`, si considera l'iterazione del *Metodo di Bisezione*:

$$f(x_k) = \frac{f(b_k) - f(a_k)}{b_k - a_k}$$

Ordine di convergenza: $p \geq 1$
Implementazione: la si può schematizzare nel seguente modo:

$$c_k = \frac{a_k + b_k}{2} \qquad per\ \ k = 1, 2, \ldots$$

$$rp = \left| \frac{f(x_k)}{f'(x_k)} \right|$$

2. Svolgimento

Questa function, nello specifico, si è implementata da zero. Una volta inserita nell'editor di MATLAB la sintassi della function `bisezione.m` (fornita), si sono inserite le indicazioni dell'implementazione tramite semplici commenti. Sulla base teorica enunciata in breve alla sottosezione 1.3 (pagina 4), si è proseguita l'implementazione basando maggior parte dell'attenzione al ciclo `while,` nel quale si sono aggiute le formule base carattestiche di qualunque metodo iterativo e le formule tipiche del metodo realizzato. Al termine di questa prima parte implementativa, si è testata la correttezza della function `bisezione.m` grazie all'editazione dello script. L'utilizzo dello script per questa tipologia di verifica ha dato la possibilità di correggere la function `bisezione.m` precedentemente implementata, dato che si ottenevano dei risultati non congrui alla teoria ed alle risultanze fornite dal docente del corso.

3. Dati, calcoli e grafici

3.1 Dati e calcoli

Sulla base della function (2), tramite lo script creato per il test dei risultati, si è stabilita la seguente equazione fornita dal testo della prova:

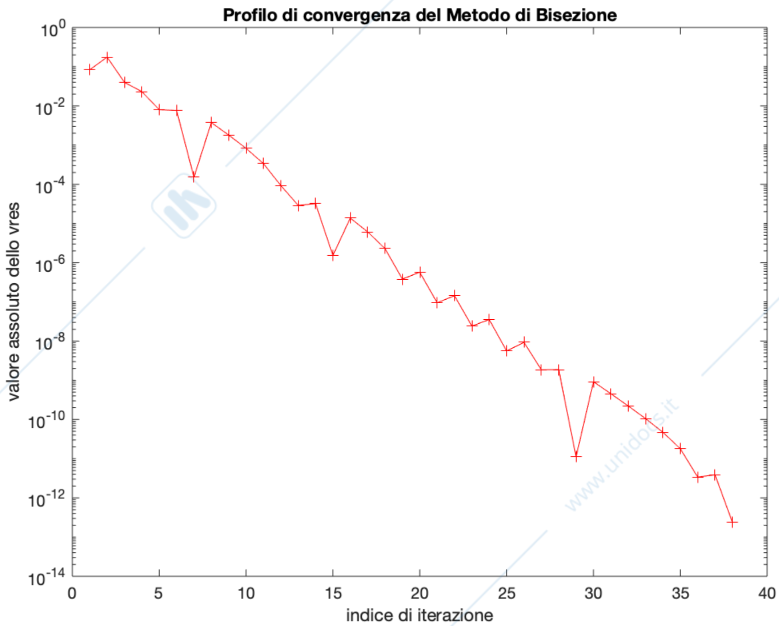
$$f(x) = x^2 - 2$$

ed i parametri forniti.
I risultati finali restituiti a video dal software MATLAB all'ultima iterazione sono i seguenti:

iter	ck	vres
38	1.414213562373	2.38811e-13

3.2 Grafici

Vista la semplicità della funzione $f(x) = x^2 - 2$ inserita nello script non risulta utile ottenere a video il grafico della funzione stessa. Interessante è, invece, osservare il profilo di convergenza del *Metodo di Bisezione* di seguito riportato:



Secondo quanto è visibile dalla tabulazione sopra, il profilo decresce in maniera più o meno lineare. Questo è un comportamento, secondo l'analisi effettuata, che rientra nei risultati teorici studiati. Si riscontra graficamente, come specificato analiticamente alla sottosezione 3.1 (pagina 5), che il *Metodo di Bisezione*, in questo specifico studio, converge in 38 iterazioni (passi).

4. Conclusioni

Il *Metodo di Bisezione* è un metodo iterativo lento, ma detiene il pregio di avere un limite superiore al numero di iterazioni necessario per ottenere un'approssimazione della soluzione. Oltre a ciò il *Metodo di Bisezione* è sempre applicabile se la funzione $f(x)$ è definita in tutto [a,b]. Inoltre converge sempre, per questo si dice che gode della cosiddetta convergenza totale.

ESERCIZIO 3

1. Introduzione

In questo esercizio, considerando il numero di matricola dello studente, sono stati implementati, attraverso il software di calcolo MATLAB, diversi metodi iterativi per la risoluzione di una data equazione $f(x) = 0$.

1.1 Testo esercizio

Si considerino le ultime 4 cifre del proprio numero di matricola con le specifiche indicate nella prova. Si consideri quindi il problema di risolvere l'equazione $f(x) = 0$, dove:

$$f(x) = 2\alpha x^3 - \beta x + \gamma e^{2x} - 2\delta$$

Si proceda ora con la stesura dello `scriptes1.m` con tutte le functions, teoremi, metodi e valori forniti nello specifico nella prova.

1.2 Scopo

Utilizzare i metodi iterativi richiesti per risolvere l'equazione relativa al numero di matricola dello studente.

1.3 Cenni teorici

- Teorema degli Zeri:

Si consideri una funzione:

$$f : Dom(f) \subseteq \mathbb{R} \longrightarrow \mathbb{R} \text{ e sia } [a, b] \subseteq Dom(f)$$

allora,

$$f(a) \cdot f(b) < 0$$

- Metodo di Bisezione:

La teoria del *Metodo di Bisezione* è enunciata in breve alla pagina 4 (sottosezione 1.3).

- Metodo di Newton-Raphson:

Si definisce `fzero` un punto ξ tale che $f(\xi) = 0$. Il metodo richiede che:

1. f sia continua e derivabile in $I = [a, b] \in \mathbb{R}$;
2. $f'(x) \neq 0 \ \forall x \in [a, b]$.

Allora, se ben definito, fissato un punto iniziale x_0, si considera l'iterazione del *Metodo di N-R*:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x)} \qquad con \ k \geq 0$$

Per il calcolo della costante di convergenza (attraverso gli scarti) e l'errore del *Metodo di N-R* valgono le seguenti formule:

$$M \sim \frac{|s_k|}{s_{k-1}^2} \quad ; \quad \epsilon_k = |(\xi - x_k)|$$

- Metodo di Steffensen:

La prima parte teorica del *Metodo di Steffensen* è enunciata in breve alla pagina 2 (sottosezione 1.3).

Per il calcolo della costante di convergenza (attraverso le derivate) e l'errore del *Metodo di Steffensen* valgono le seguenti formule:

$$M = \left|\frac{1+f'(\xi)}{f'(\xi)}\right| \frac{|f''(\xi)|}{2} \quad ; \quad \epsilon_k = |(\xi - x_k)|$$

- Metodo della Tangente Fissa:

Per il calcolo della costante di convergenza (attraverso le derivate) e della stima delle iterazioni del *Metodo della Tangente Fissa* valgono le seguenti formule:

$$M = \left|\frac{1-f'(\xi)}{x_0}\right| \quad ; \quad k \simeq \frac{log_{10}(|tol|)}{sgn[log_{10}(M)]} \quad con\ sgn[z] = \begin{cases} z & \text{se } tol \geq 1 \\ -z & \text{se } tol < 1 \end{cases}$$

2. Svolgimento

A seguito della creazione su MATLAB di uno script denominato `scriptes1.m`, si è iniziata la stesura del programma dall'editor del software secondo l'ordine preposto dalla prova (a⇝g). Una volta definito il numero personale di matricola, si è richiamata la function `param.m`, la quale secondo le denominazioni definite aumenta e/o divide le ultime 4 cifre della matricola [2]. Allorché sono state inserite e calcolate la $f(x)$, la $f'(x)$ e la $f''(x)$[3]; si è proseguito con il definire, sulla base teorica enunciata in breve alla sottosezione 1.3 (pagine 6-7), il *Teorema degli Zeri* e con il richiamare (in ordine di utilizzo pratico) le functions `bisezione.m`, `newton.m`, `steffensen.m` ed i loro relativi errori finali e costanti del fattore di convergenza. Queste ultime, solo in corso di progettazione informatica, sono state stimate attraverso sia l'utilizzo delle derivate, sia tramite gli scarti per poter avere un controllo capillare dello script. Per quando concerne il *Metodo della Tangente Fissa*, è stata prima stimata la sua costante asintotica e successivamente stimato il suo numero di iterazioni secondo quanto esposto in seguito. In corso d'opera si sono per di più aggiunti i grafici analizzati alle pagine 8 e 9 ed infine il comando per la creazione di un file di testo `output.txt` per il salvataggio dei risultati.

3. Dati, calcoli e grafici

Numerosi sono i dati inseriti ed i calcoli eseguiti dal software MATLAB.

3.1 Dati e calcoli

Come primo approccio allo script la function `param.m`, la cui sintassi è la seguente:

```
[alfa,beta,gamma,delta] = param(matricola)
```
(3)

ha calcolato, secondo quanto definito sul testo della prova, i seguenti valori[4]:

α	β	γ	δ
10	10	0.10	4

[2] Nel caso personale studiato essendo il numero di matricola pari a 1219903, le ultime 4 cifre sono le seguenti: 9903.
[3] Su MATLAB esse sono identificare come: `fun, dfun, d2fun`.
[4] I valori espressi nella tabella variano al variare del numero di matricola definito all'inizio dello script.

Successivamente, sempre sulla base teorica presente alle pagine 6-7 (sottosezione 1.3), il software ha calcolato tramite le functions richiamate nello script i risultati inseriti in seguito.
Attraverso la function (2) `bisezione.m`, MATLAB ha elaborato un valore di punto iniziale necessario pari a `x0 = 2.0(= cnew)`. Questo valore è divenuto il punto iniziale x_0 necessario alla function `newton.m`, con la seguente sintassi:

`[iterN,xnew,scartiN] = newton(x0,tolN,itmaxN,fun,dfun)` (4)

Proprio attraverso la function (4) (ed altre formule teoriche presenti sopra), si sono elaborati i seguenti valori:

xk(NR)	iter(NR)	ϵ(NR)	MNR
0.942988123677	7	0.00000000e+00	1.2962

In corso di analisi critica dei risultati è apparso particolare il risultato dell'errore finale del *Metodo di Newton-R.* che è pari a 0 (zero)[5]. Questa soluzione può comunque essere valida e quindi corretta visto che ϵ_k, anche da un punto di vista generico, potrebbe essere meno della precisione di macchina[6].
Proseguendo si è richiamata la function (1) `steffensen.m` e le sue relative formule espresse sopra, da cui si sono ottenuti i seguenti risultati:

xk(S)	iter(S)	ϵ(S)	MS
0.942988123678	5	5.64437386e-13	59.1939

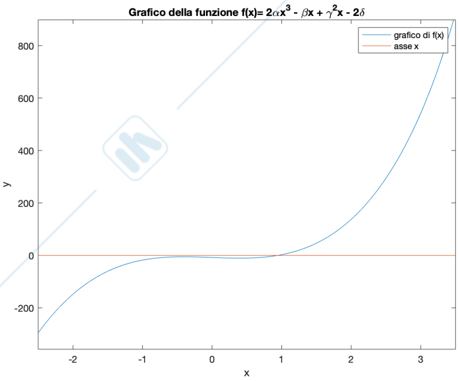
Tutti i valori risultano essere corretti e, conoscendo la teoria dell'argomento, non è stata necessaria la prosecuzione dell'analisi critica dei risultati. Si noti, comunque, che il risultato della costante asintotica del *Metodo di Steffensen* (`MS`), risulta essere molto precisato dato che, rispetto a quella del *Metodo di N-R* è stata ottenuta tramite la formula con le derivate della funzione e non attraverso gli scarti.
L'ultimo risultato ottenuto dallo `scriptes1.m` è la stima delle iterazioni con il *Metodo di Tangente Fissa* tramite il suo valore della costante asintotica.

$\tilde{\texttt{iter}}$(TF)
89.82

Questo valore, quindi, essendo un'iterazione e quindi un numero solitamente intero, può essere approssimato a 90. Nel caso in cui il *Metodo di Tangente Fissa* non convergeva $\tilde{\texttt{iter}}$`(TF) = 100`.

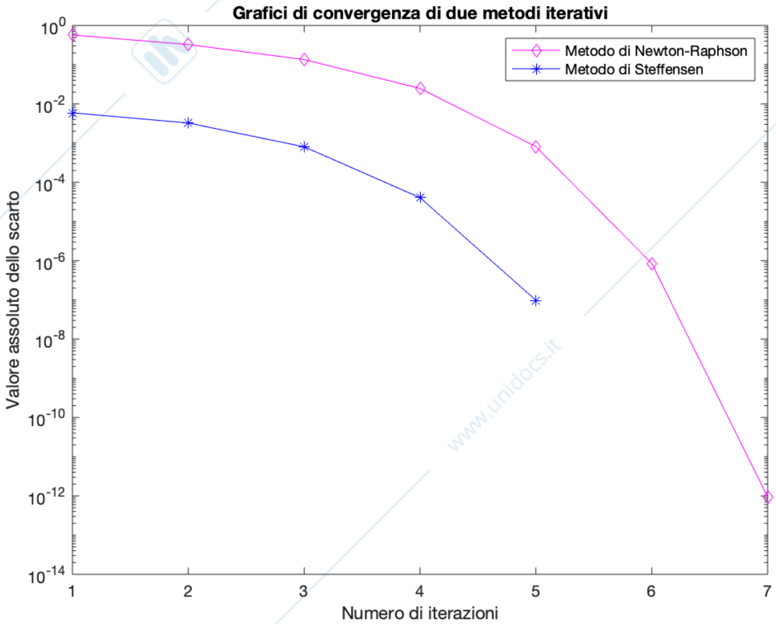
3.2 Grafici

Ai fini della relazione risulta interessante analizzare i due grafici ottenuti a video dallo `scriptes1.m`. Secondo i valori calcolati di $\alpha, \beta, \gamma, \delta$ presenti nella tabella al termine della pagina 7, è stata calcolata la (personale) funzione $f(x)$, che in maniera grafica è rappresentata come segue:



[5] Se pur viene visualizzato un risultato ϵ_k `= 0`, questo non è realmente pari a zero ma pari ad un numero j$\in \mathbb{R}$ non visualizzabile su MATLAB.
[6] In MATLAB la precisione di macchina risulta essere `eps = 2.220446049250313e-16`.

Per quanto concerne la seconda figura ottenuta in MATLAB, si ha il grafico di convergenza dei due metodi iterativi principali presenti nello `scriptes1.m`, ovvero il *Metodo di Newton-Raphson e di Steffensen.*



Dall'antecedente si può vedere come il *Metodo di Steffensen* converge ad un minor numero di iterazioni (nello specifico, 4 passi), mentre il *Metodo di N-R* risulta più lento con 2 iterazioni in più rispetto all'altro metodo. Da questo grafico, comunque, non è possibile definire quale dei due metodi è maggiormente accurato, dato che ciò è definibile solamente basandosi sugli errori finali o veri dei rispettivi metodi ed applicando un confronto fra di essi.

4. Conclusioni

Dall'esercizio svolto siamo stati in grado di analizzare da un punto di vista pratico tramite MATLAB l'accuratezza e la velocità alla quale il software di calcolo numerico ci fornisce determinati risultati. Dai risultati ottenuti, secondo l'equazione $f(x) = 0$ studiata, si può dichiarare con certezza che il *Metodo di Newton-Raphson*, nonostante converga in 7 iterazioni, è più accurato rispetto al *Metodo di Steffensen* (convergenza in 5 passi). Dal primo metodo, infatti, risulta un errore finale ϵ`(NR) = 0`, mentre nel secondo ϵ`(S) = 5.64437386e-13`.

ESERCIZIO 4

1. Introduzione

In questo esercizio è stato implementato, attraverso il software di calcolo MATLAB, il *Metodo di Aitken*, sulla base del *Metodo di Punto Fisso* studiato a lezione.

1.1 Testo esercizio

Si scriva una function MATLAB che implementi il *Metodo di estrapolazione Aitken*. Tale function dovrà avere come parametri di inputs gli stessi della function che implementa il punto fisso.

1.2 Scopo

Utilizzare la function `aitken.m` implementata per la soluzione iterativa di una generica equazione non lineare $g(x) = 0$ con il *Metodo di Aitken*.

1.3 Cenni teorici

Per risolvere $g(x) = 0$, fissato il punto iniziale x_0, si considera l'iterazione del *Metodo di Aitken*:

$$x_{k+1} = x_k - \frac{(g(x_k) - x_k)^2}{g(g(x_k)) - 2g(x_k) + x_k}$$

Ordine di convergenza: $p = 2$
Implementazione: applicata al *Metodo di Punto Fisso* la si può schematizzare nel seguente modo:

$$x_1 = x_k$$

$$x_2 = g(x_1)$$

$$x_3 = g(x_2)$$

$$x_{k+1} = x_1 - \frac{(x_2 - x_1)^2}{x_3 - 2x_2 + x_1}$$

2. Svolgimento

Visti i nessi fra *Metodo di Punto Fisso* e quello di *Metodo di Aitken*, per implementare su MATLAB la function `aitken.m` richiesta, si è copiata la function `pfisso.m` studiata nel corso delle lezioni telematiche e si sono svolti alcuni cambiamenti tramite l'editor del software. Seguendo, come precedentemente, la teoria del *Metodo di Aitken* riassunta alla sottosezione 1.3 (pagina 7), si è conclusa la prima parte dell'esercitazione. Per controllare la correttezza dello function `aitken.m`, si è deciso di utilizzare uno script, simile a quello utilizzato per testare la function `pfisso.m`.

3. Dati, calcoli e grafici

La sintassi della function implementata nell'editor di MATLAB è la seguente:

```
[xnewA,iterA,scartiA] = aitken(g,x0,tolA,itmaxA)        (5)
```

3.1 Dati e calcoli

Sulla base della function (5), tramite lo script di controllo sviluppato, si è scelta come esempio la funzione:

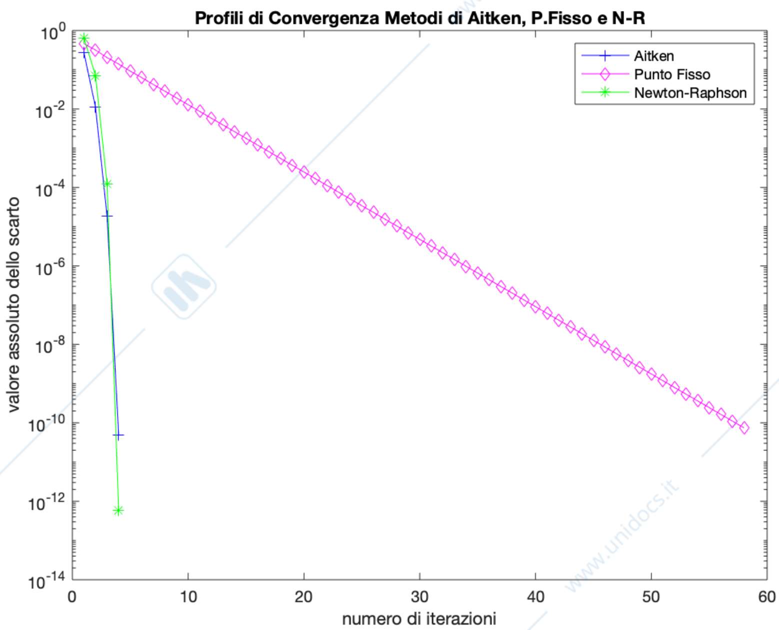
$$g(x) = cos(x)$$

ed i parametri [x0 = 1; tolA = 1e-10; itmaxA = 100].
I risultati all'ultima iterazione, congrui alla teoria, sono i seguenti:

iter(A)	x1	x2	x3	|sk|(A)
4	0.739085133166	0.739085133248	0.739085133193	4.90850e-11

3.2 Grafici

Data la semplicità della funzione $g(x) = cos(x)$ presa come esempio, non è necessario visualizzare il suo grafico. Risulta però necessario per l'analisi dell'esercizio la visualizzazione del grafico semilogaritmico di convergenza del *Metodo di Aitken* confrontato con altri due metodi iterativi.



Per analizzare, quindi, al meglio il *Metodo di Aitken* può essere utile metterlo a confronto con i *Metodi di Punto Fisso* e *Newton-Raphson*. Come si può osservare dal grafico soprastante il *Metodo di N-R* restituisce, a meno degli scarti in valore assoluto, pari valori di iterazioni al *Metodo di Aitken*. Il *Metodo di Punto Fisso*, invece, risulta visivamente differente, vista la sua caratteristica, anche teorica, di convergenza lineare (lenta). Da questo grafico, comunque, non è possibile definire quale dei metodi risulta maggiormente accurato dato che non siamo in possesso dell'errore dei rispettivi metodi.

4. Conclusioni

Il *Metodo di Aitken* è un metodo utilizzato come soluzione alla problematica presentata dal *Metodo di Punto Fisso*, ovvero dell'essere lento (convergenza lineare). Per questa motivazione il metodo studiato in questo esercizio viene anche definito *Metodo di accelerazione (o estrapolazione) di Aitken* proprio per la sua velocità di fornire valori degli scarti eguagliabili al *Metodo di P. Fisso*, ma con un minor numero di iterazioni.

ESERCIZIO 5

1. Introduzione

Nell'ultimo esercizio proposto, considerando il numero di matricola dello studente, sono stati implementati, tramite il software di calcolo MATLAB, alcuni metodi di iterazione per la risoluzione delle date funzioni $f(x)$ e $g(x)$.

1.1 Testo esercizio

Per risolvere l'equazione data all'esercizio 3 $f(x) = 0$, si usi il seguente metodo di punto fisso:

$$x_{k+1} = \frac{2\alpha x_k^3 + \gamma e^{2x_k} - 2\delta}{\beta} \equiv g(x_k)$$

Si proceda ora con la stesura dello `scriptes2.m` con tutte le functions, metodi e valori forniti nello specifico nel testo della prova.

1.2 Scopo

Utilizzare i metodi di iterazione richiesti per risolvere l'equazione relativa al numero di matricola dello studente.

1.3 Cenni teorici

- Metodo Newton-Raphson:

La prima parte teoria del *Metodo di N-R* è enunciata in breve alla pagina 6 (sottosezione 1.3).

- Metodo di Punto Fisso:

Si pone come segue:

$$g(x) = x + f(x)$$

per avere

$$g(x) = x \Leftrightarrow f(x) = 0$$

Per il calcolo della costante asitotica (attraverso la derivata) del *Metodo di P. Fisso* vale la seguente formula[7]:

$$M = \left|g'(\xi)\right| < 1 \quad e \quad M \neq 0$$

- Metodo di Aitken:

La prima parte teoria del *Metodo di Aitklen* è enunciata in breve alla pagina 10 (sottosezione 1.3).

Per il calcolo della costante di convergenza (attraverso gli scarti) e l'errore del *Metodo di Aitken* valgono le seguenti formule:

$$M \sim \frac{|s_k|}{s_{k-1}^2} \quad ; \quad \epsilon_k = |(\xi - x_k)|$$

[7] La formula $M = |g'(\xi)|$ è valida $\Leftrightarrow p = 1$ come nel caso analizzato nell'esercizio.

2. Svolgimento

A seguito della realizzazione su software MATLAB dello script denominato `scriptes2.m`, si è iniziata la stesura sull'editor secondo l'ordine a⇝c dell'esercizio proposto. Una volta definito il numero di matricola personale e richiamata la function `param.m`, come nell'esercizio 3, si sono inserite e calcolate le funzioni $f(x), f'(x), g(x)$ e $g'(x)$[8]. Successivamente, sulla base teorica riassunta alla sottosezione 1.3 (pagina 12), si sono richiamate le functions (in ordine) `newton.m`, `pfisso.m`, `aitken.m` e definite ulteriori formule note per il calcolo dei valori richiesti dalla prova. Infine si è realizzato un grafico in scala semilogaritmica e si sono aggiunti automaticamente al file di testo `output.txt`, precedentemente creato, i risultati ottenuti dallo script dell'esercizio.

3. Dati, calcoli e grafici

3.1 Dati e calcoli

Come primo approccio allo script si è richiamata la function (3) `param.m`, che ha fornito gli stessi valori di $\alpha, \beta, \gamma, \delta$ scritti nella tabella posta alla fine della pagina 7.
In secondo luogo, visto che il testo della prova richiedeva di utilizzare come soluzione vera la `xnew` del *Metodo di N-R*, per generalizzare il tutto a qualsiasi numero di matricola è stata inserira la function (4) `newton.m`, alla quale si sono stati immessi come inputs, oltre che la funzione `(fun)` e la sua derivata prima, i seguenti parametri: `[x0 = 1; tolN = 1e-08; itmaxN = 30]`. Questi tre parametri sono stati posti in maniera arbitraria, scegliendo, nello specifico, un punto iniziale che rientrasse nell'intervallo di studio della funzione (personale). Tramite questa function si sono ottenuti i seguenti risultati finali:

iter(NR)	xk(NR)	|sk|(NR)
4	0.942988123677	4.50525e-10

Successivamente, calcolata la costante asintotica del punto fisso (come definito dalla teoria), tramite un ciclo `if` si è richiamata la function `pfisso.m`, la cui sintesi è la seguente:

$$\texttt{[xnewP,iterP, scartiP] = pfisso(g,x0,tolP,itmaxP)} \tag{6}$$

Dipoi, studiata la divergenza[9] del *Metodo di Punto Fisso*, si è richiamata la function (5) `aitken.m`, la quale attraverso il calcolatore, ci ha permesso di visualizzare i seguenti risultati finali:

xk(A)	iter(A)	ϵ(A)	MA
0.942988123677	4	1.11022302e-16	7.0860

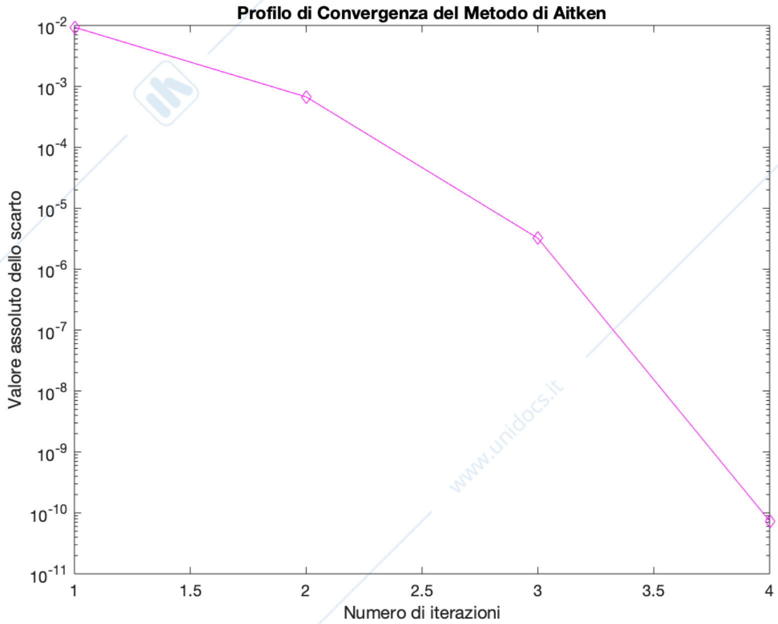
I valori ottenuti risultano essere, anche basandosi sulla teoria dell'argomenti, corretti. Si noti che l'errore finale con il *Metodo di Aitken* è abbastanza vicino alla precisione di macchina (per il valore `eps` vedi nota *5*).

3.2 Grafici

Secondo quanto è stato richiesto dalla prova si è ottenuta una visualizzazione a video del profilo di convergenza del *Metodo di Aitken*. Visto che, il *Metodo di Punto Fisso* diverge non si sono messi a confronto i due profili dei rispettivi metodi. Questa esclusione (o meno) del profilo semilogaritmico di Punto Fisso è stata possibile attraverso un ciclo `if`.

[8] Su MATLAB esse sono identificare come: `fun, dfun, g, dg`.
[9] Il *Metodo di Punto Fisso* converge se $-1 < M < 1$, mentre diverge se $M < -1 \bigcup M > 1$.

Per le ragioni esposte, ne segue, quindi, la figura del profilo del *Metodo di Aitken*.



Dal grafico al di sopra si confermano i risultati finali presentati sotto forma di tabella alla pagina 13.

4. Conclusioni

Secondo quanto si è notato dai risultati analitici e grafici ottenuti tramite lo `scriptes2.m`, il *Metodo di Punto Fisso*, che, come si è dimostrato nell'esercizio 4, risulta lento e diverge. Ciò è spiegabile dal fatto che il suo fattore di convergenze risulta essere un numero > 1, infatti `MPF` $\sim$ `4.6672`. Efficace è, invece, il *Metodo di Aitken* che oltre ad essere veloce (chiamato infatti di accelerazione) risulta anche accurato visto il suo valore di errore finale è pari a ϵ`(A) = 1.11022302e-16` e quindi prossimo alla precisione di macchina.

Autore:
La presente Relazione Finale è stata interamente svolta dello studente Oliveri Marco (n. matr. 1219903) regolarmente iscritto al Corso di Laurea in Ingegneria Aerospaziale (Canale B) del Dipartimento di Ingegneria Industriale DII.
Per qualsiasi comunicazione all'autore inviare una email all'indirizzo *marco.oliveri@studenti.unipd.it*.

Realizzazione:
La realizzazione grafica, la composizione e l'impaginazione della Relazione Finale è stata eseguita attraverso l'editor TEXMAKER con linguaggio LaTeX. La realizzazione dei grafici allegati sono stati eseguiti attraverso il software MATLAB. Coordinamento, redazione e ricerca eseguita interamente dall'autore della presente relazione. I programmi principali citati ed ogni altro programma secondario utilizzati in corso d'opera sono stati scaricati con relative licenze su dispositivi elettronici di produzione Apple (MacBook Pro 13-inch, 2019, Four Thunderbolt 3 ports e iMac - 21.5-inch, 2008).

UNIVERSITÀ DEGLI STUDI DI PADOVA