Metodi Numerici per il Calcolo

Esercitazione 3: Script, Function e Numeri Finiti

A.A.2022/23

Scaricare dalla pagina web del corso l'archivio matlab_mnc2223_3.zip e scompattarlo nella propria home directory. Verrà creata una cartella con lo stesso nome contenente alcuni semplici script e function Matlab/Octave. Si svolga la seguente esercitazione che ha come obiettivo approfondire la propria conoscenza dell'ambiente Matlab capendo, usando e modificando piccoli script e function sulla rappresentazione e aritmetica floating point (Numeri Finiti). Gli script proposti vogliono mettere in pratica alcuni concetti visti a lezione; per rafforzare la propria comprensione i vari script vanno modificati e rieseguiti più volte.

Precisione BASIC single e BASIC double
Matlab usa di default la precisione double e si può passare in precisione single mediante l'utilizzo della funzione `single()`. Poter alternare in un codice le precisioni single e double può essere molto utile a fini didattici.

A. Script `scompute_u.m`
Lo script in oggetto calcola l'unità di arrotondamento U, sia in precisione single che double, mediante la seguente definizione operativa (ANSI/IEEE std.754):
"U è il più grande numero finito positivo tale che $fl(U + 1) = 1$".
Analizzare l'implementazione della definzione operativa e verificare i risultati prodotti nei casi single e double.

B. Ciclo while e numeri gradual underflow; script `sfiniti.m`
Analizzare lo script che implementa un piccolo ciclo sia in versione BASIC single che BASIC double (vedi commenti) e stampa ad ogni iterazione un numero finito.

1. Prima di eseguire lo script prevedere cosa verrà prodotto in stampa.
2. Eseguire lo script, analizzare i risultati e individuare i numeri finiti "gradual underflow" dello standard ANSI/IEEE (aiutarsi con la WebApp IEEE-754 Floating-Point Conversion).
3. Modificare la condizione del ciclo `while` da `x>0` all'equivalente `x+1>1`; l'output rimane lo stesso'? Spiegare cosa avviene.

C. Script `sexpression.m`
Viene effettuato il calcolo della semplice espressione

$$y = ((1 + x) - 1)/x$$

che dovrebbe sempre produrre come risultato il valore esatto 1, invece a seconda del valore assegnato ad x si ottengono risultati inattesi. Provare differenti valori per x e dedurre per quali il risultato sarà corretto e per quali no. Usufruire dello script `sconv_dec2bin.m` che permette di convertire un numero decimale in base 2 e controlla se è rappresentabile in modo esatto.

D. Approssimazione della derivata; script `fidiff.m`

Sia f una funzione continua e derivabile e cerchiamo di approssimare il valore della derivata mediante il limite del rapporto incrementale

$$\lim_{h \to 0} \frac{f(x + h) - f(x)}{h}.$$

L'idea è di valutare il rapporto incrementale per un certo valore di h, ma quanto piccolo? Ponendo $h = 0$ si ottiene $0/0$, cioè NaN. Lo script in oggetto prova valori di h da 10^0 a 10^{-14} assumendo come x il valore 1.0, se non ne viene fornito uno differente.

Lo script fa uso di precisione double. Si vede che l'approssimazione migliora al diminuire di h, ma quando h diventa troppo piccolo, l'approssimazione comincia a peggiorare. Spiegarne il perché.

Osserviamo meglio l'output numerico; si nota che per valori grandi di h, l'errore assoluto diminuisce di un fattore 10 ogni volta che h viene ridotto di un fattore 10, fino a che poi si ottengono risultati inattendibili. Spiegare il perché.

E. Sull'Errore Inerentei nella valutazione di una funzione
L'errore inerente misura l'accuratezza che ci si può aspettare nel risolvere un problema in una data precisione, indipendentemente dall'algoritmo usato. Se consideriamo il problema di valutare una funzione reale di variabile reale derivabile almeno due volte avremo la seguente stima del numero di condizione di f in x:

$$C(f, x) = \frac{|xf'(x)|}{|f(x)|}.$$

Quante cifre esatte del risultato ci aspettiamo nel valutare $f(x)$? Ricordando la stima dell'Errore Inerente

$$\frac{|f(\tilde{x}) - f(x)|}{|f(x)|} \approx C(f, x) \frac{|\tilde{x} - x|}{|x|}$$

e applicando il logaritmo in base 10 e poi cambiando segno si ha:

$$-\log_{10}\left(\frac{|f(\tilde{x}) - f(x)|}{|f(x)|}\right) \approx -\log_{10}\left(\frac{|\tilde{x} - x|}{|x|}\right) - \log_{10}(C(f, x)).$$

Il membro di sinistra rappresenta il numero di cifre esatte con cui $f(\tilde{x})$ approssima $f(x)$, così come il primo termine del membro di destra rappresenta il numero di cifre esatte con cui $\tilde{x}$ approssima x e che sappiamo quante sono a seconda della precisione di rappresentazione usata. Quindi la stima del numero di cifre che si ottengono nel risultato si ottiene sottraendo la quantità $\log_{10}(C(f,x))$ al numero di cifre corrispondente alla precisione usata.

Se consideriamo la funzione $\sin(x)$, teoricamente si ottiene:

f	x	$f(x)$	$f'(x)$	$C(f,x)$	$\log_{10}(C(f,x))$
sin	π	0	-1	∞	∞
sin	$\pi/2$	1	0	0	$-\infty$
sin	0	0	1	NaN	NaN

Si consideri lo script `scond_sin.m` che calcola l'Errore Inerente tra il valore calcolato in doppia precisione (che assumeremo come risultato esatto a partire da un dato esatto) ed il valore calcolato in precisione singola (che assumeremo come risultato esatto, ma a partire da un dato perturbato) per sperimentare se numericamente si ha un comportamento in linea con quanto previsto teoricamente.
Si modifichi lo script per provare i valori indicati in tabella, con $eps1 = 2^{-12}$ e $eps2 = 2^{-24}$. Dai risultati ottenuti che considerazioni si possono fare?

f	x double	$f(x)$, x single	$f(x)$, x double	$-\log_{10}(Ein)$	$\log_{10}(C(f,x))$
sin	π				
sin	$\pi + eps1$				
sin	$\pi + eps2$				
sin	$\pi/2$				
sin	$\pi/2 + eps1$				
sin	$\pi/2 + eps2$				
sin	$eps1$				
sin	$eps2$				

F. Esercizio di verifica
Nella function `compound.m` viene implementata, in BASIC single, la formula dell'interesse composto.

Supponiamo di investire a_0 euro in una banca che paga il 5% di interesse composto a trimestre; questo significa che alla fine del primo trimestre dell'anno, il valore dell'investimento è:

$$a_1 = a_0(1 + 0.05/4) \qquad euro$$

cioè l'ammontare originale più 1/4 del 5% della somma iniziale. Alla fine del secondo trimestre, la banca paga l'interesse non solo su a_0, ma anche sull'interesse maturato nel primo trimestre. Così il valore dell'investimento alla fine del secondo trimestre è:

$$a_2 = a_1(1 + 0.05/4) = a_0(1 + 0.05/4)^2 \qquad euro.$$

Alla fine del terzo trimestre la banca paga:

$$a_3 = a_2(1 + 0.05/4) = a_0(1 + 0.05/4)^3 \qquad euro,$$

e alla fine dell'anno paga

$$a_4 = a_0(1 + 0.05/4)^4 \qquad euro.$$

In generale, se si investe a_0 euro ad un interesse composto x, n volte all'anno, alla fine dell'anno il valore è $a_0 C_n(x)$ euro, dove

$$C_n(x) = (1 + \frac{x}{n})^n.$$

Prima di considerare un algoritmo, analizziamo il numero di condizione del problema di calcolare $C_n(x)$. Si ha

$$C_n'(x) = n(1 + \frac{x}{n})^{n-1}\frac{1}{n} = \frac{C_n(x)}{1 + \frac{x}{n}}.$$

Così, per n sufficientemente grande rispetto ad $|x|$, $C_n(x)$ approssima la sua derivata, che non è sorprendente, infatti è noto che fissato x, $C_n(x)$ ha come limite per $n \to \infty$, esattamente e^x, e la derivata di e^x è se stessa.

Perciò, il numero di condizione di $C_n(x)$ è

$$C(C_n, x) = \frac{|x|}{|C_n(x)|}\frac{C_n(x)}{1 + \frac{x}{n}} = \frac{|x|}{1 + \frac{x}{n}},$$

che converge ad $|x|$, per $n \to \infty$. Si noti che $|x|$ è anche il numero di condizione di e^x. Come conseguenza, il problema del calcolo dell'interesse composto è ben condizionato anche per valori di n molto grandi, quando $|x|$ non è grande.

La function `compound` implementa cinque differenti algoritmi per calcolare l'interesse composto e produce una stampa con i risultati. Si modifichi lo script per trasformarla in una function (la si chiami `fcompound.m`) che ritorni i risultati ottenuti, quindi si realizzi uno script (lo si chiami `scompound.m`) che richiami la function e produca una tabella simile alla seguente.

n/Alg	** Alg1 **	** Alg2 **	** Alg3 **	** Alg4 **	** Alg5 **
50					
100					
1000					
10000					
100000					
1000000					

Per quanto spiegato, aumentando n, il valore deve tendere ad e^x. I risultati ottenuti con i differenti Algoritmi non sono tutti uguali fra loro; quali sono i corretti? Come ci si spiega quanto ottenuto?