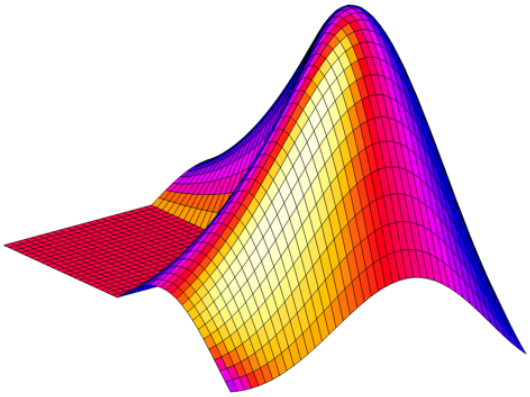
1 Introduzione

MATLAB è un ambiente interattivo di alto livello per il **calcolo scientifico**, scritto soprattutto in linguaggio `C` e `C++`. Esso permette la soluzione numerica ad alte prestazioni di problemi tramite la *programmazione* e *visualizzazione grafica*. L'acronimo MATLAB sta per **MATrix LABoratory**, in quanto le sue unità di base sono dati di tipo matriciali. Esso è stato infatti originariamente sviluppato destinato alla gestione e sviluppo di calcoli matriciali. MATLAB è distribuito da `The MathWorks` (www.mathworks.com), generalmente a pagamento. Octave (www.octave.org) è un interprete per linguaggio di alto livello largamente compatibile con MATLAB, distribuito gratuitamente, che riproduce una larga parte dei comandi MATLAB.



Entrambi gli ambienti sono caratterizzati da dei pacchetti aggiuntivi, chiamati **toolbox** in MATLAB e **packages** in Octave. Questi pacchetti sono collezioni di programmi e algoritmi relativi ad uno specifico argomento, scritti sotto file speciali detti M-file oppure file con estensione .m.

1.1 Ambiente di lavoro

Dopo aver installato MATLAB sul proprio computer, quando lanciamo il file eseguibile si apre un desktop di lavoro suddiviso in sottofinestre. La finestra principale è la finestra di comando (**Command Window**), dove si digitano i comandi e le istruzioni. Essa è caratterizzata dalla presenza del **promt**

```
>>
```

che rappresenta la *linea di comando* di MATLAB che indica che MATLAB è pronto ad accettare comandi, quali dichiarazioni di variabili, espressioni e chiamate alle funzioni presenti. La sottofinestra denominata **Workspace** contiene le variabili che sono state dichiarate. Esistono dei comandi specifici per accedere e manipolare lo spazio di lavoro, eseguibili sulla linea di comando

`who`	nome delle variabili in uso
`whos`	nome, dimensioni e tipo di dato delle variabili in uso
`clear`	elimina tutte le variabili dal Workspace
`clear a`	elimina solo la variabile `a` dal Workspace
`save`	salva in un MAT-file il contenuto del Workspace
`save a`	salva in un MAT-file il contenuto della variabile `a`
`load`	ricarica le variabili salvate tramite `save`
`load a`	ricarica la variabile `a` salvata tramite `save a`

Tra i comandi più importanti di MATLAB, troviamo il comando `help`, che permette di accedere a tutte le infomazioni che riguardano la sintassi di una funzione o istruzione richiesta

```
help istruzione  (oppure funzione)
```

Quando non si conosce il nome esatto del comando si può fare riferimento al comando `lookfor`, che opera tramite dei *keywords*. Un'altro comando di uso generale è `clc` che pulisce la Command Window, ossia lo schermo, ma non pulisce la memoria del Workspace. Inoltre, premendo i tasti `'Ctrl c'` facciamo terminare un programma che sta girando. Per uscire da MATLAB si digita il comando `quit` sulla linea di comando.

Il comando `=` viene utilizzato per *dichiarare* una variabile assegnandole un valore

```
x=3 %dichiariamo la variabile x assegnandole il valore 3
```

MATLAB è **case-sensitive**, che significa che distingue tra maiuscole e minuscole. Di conseguenza, le variabili `b` e `B` sono distinte tra loro. Alcune regole generali per le variabili sono che accetta nomi di variabili lunghi fino ad un massimo di 19 caratteri alfanumerici, e il primo deve essere una lettera. In generale, ogni istruzione va terminata da un punto e virgola `;` in modo da evitare che venga visualizzato il suo contenuto sullo schermo. In caso contratio, non si mette il punto e virgola.

2 Aritmetica di macchina

La maggior parte dei problemi della vita reale **non** possono essere risolti in maniera *analitica esatta*, bensì solamente in maniera **approssimata** tramite l'uso di una **macchina** (o **calcolatore**), le cui risorse sono necessariamente **finite**.

In particolare, un calcolatore non può memorizzare, di conseguenza neanche rappresentare, l'infinità dei numeri reali. L'insieme $\mathbb{R}$ dei numeri reali viene approssimato da un calcolatore da un suo sottoinsieme **finito** e **discreto** chiamato l'insieme dei numeri **numeri di macchina** e indicato con $\mathbb{F}$. Ogni numero reale $x \in \mathbb{R}$ viene rappresentato dal corrispondente numero di macchina $fl(x) \in \mathbb{F}$, non necessariamente coincidente con il numero x di partenza.

Nel formato interno dell'elaboratore, i numeri di macchina sono memorizzati nel seguente formato

$$x = (-1)^s \cdot (0.a_1a_2 \dots a_t) \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-t}, \quad a_1 \neq 0, \tag{1}$$

dove s rappresenta il **segno**, che può valere 0 o 1, β è un numero intero positivo maggiore od uguale a 2 e rappresenta la **base**, m è un intero detto **mantissa** ed e è un numero intero che rappresenta l'**esponente**. Le cifre $a_1a_2 \dots a_p$ (con $p \leq t$ e $0 \leq a_i \leq \beta - 1$) sono chiamate le prime p cifre significative di x. La lunghezza della mantissa t è il numero massimo di cifre significative.

I numeri di macchina nel formato (1) sono detti numeri **floating-point normalizzati** a causa della posizione variabile del punto decimale. La condizione $a_1 \neq 0$ garantisce che un numero non abbia più rappresentazioni. Ad esempio, se questa condizione fosse violata, $1/10$ in base 10 potrebbe essere rappresentato come $0.1 \cdot 10^0$ o $0.01 \cdot 10^1$ e così via. L'insieme $\mathbb{F} = \mathbb{F}(\beta, t, L, U)$ è dunque l'insieme dei numeri floating point nel formato (1) ed è completamente caratterizzato dalla base β, dal numero di cifre significative t e dall'intervallo (L, U) (con $L < 0$ ed $U > 0$) di variabilità dell'esponente e.

MATLAB e Octave utilizzano $\mathbb{F}(2, 53, -1021, 1024)$. Questo corrisponde ad una rappresentazione dei numeri in formato con 16 cifre decimali in base 10. Il sistema ha diversi *formati di output* per rappresentare i numeri. Di fatto, lo stesso numero assume espressioni diverse se fatto precedere da opportune dichiarazioni di formato, come si può vedere nel seguente esempio per il numero razionale $x = 2/7$

formato	rappresentazione
`format short`	`0.2857`
`format short e`	`2.8571e-01`
`format short g`	`0.28571`
`format long`	`0.285714285714286`
`format long e`	`2.857142857142857e-01`
`format long g`	`0.285714285714286`

Ogniqualvolta si sostituisce un numero reale $x \neq 0$ con il suo rappresentante $fl(x) \in \mathbb{F}$, si commette un inevitabile **errore di arrotondamento**. Questo errore è generalmente piccolo e soddisfa

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\epsilon_M, \tag{2}$$

dove $\epsilon_M = \beta^{1-t}$, detta **epsilon macchina**, rappresenta la distanza fra 1 ed il più piccolo numero floating-point maggiore di 1. Si osservi che ϵ_M dipende da β e da t. In MATLAB e Octave,

$$\epsilon_M = 2^{-52} \approx 2.22 \cdot 10^{-16} \tag{3}$$

Il numero

$$u = \frac{1}{2}\epsilon_M, \tag{4}$$

viene chiamato **unità di arrotondamento** in quanto rappresenta il massimo errore relativo che la macchina può commettere nella rappresentazione di un numero reale. ϵ_M in

MATLAB e Octave è calcolabile tramite il commando `eps`, il cui valore nel formato `short` è `eps=2.2204e-16`.

Essendo gli estremi L ed U dell'intervallo di variabilità dell'esponente finiti non si potranno rappresentare numeri in valore assoluto arbitrariamente piccoli o grandi. Di fatto, il più piccolo ed il più grande numero positivo di $\mathbb{F}$ sono

$$x_{min} = \beta^{L-1} \quad x_{max} = \beta^{U}(1 - \beta^{-t}). \tag{5}$$

In MATLAB e Octave essi si determinano usando i comandi `realmin` e `realmax` e valgono `realmin=2.2251e-308` e `realmax=1.7977e+308`. Un numero positivo più piccolo di `realmin` produce una segnalazione di **underflow** e viene generalmente trattato come 0. Un numero positivo più grande di `realmax` produce una segnalazione di **overflow** e viene memorizzato nella predefinita `Inf` che rappresenta l'*infinito positivo* per il calcolatore. Un'altra variabile predefinita è la variabile `NaN` (*not a number*), che corrisponde alle forme indeterminate quali $0/0$, ∞/∞, $\infty - \infty$, le quali non possono trovare posto in $\mathbb{F}$.

3 Matrici e vettori

Come suggerisce il suo acronomo, l'elemento fondamentale di calcolo di MATLAB è la matrice. Esso è dunque basato sull'utilizzo e gestione di dati di tipo matriciale, in una o più dimensioni. Infatti in MATLAB ogni variabile è una matrice. I vettori e gli scalari sono considerati particolari matrici. L'inserimento di una matrice avviene utilizzando le parantesi quadre, separando gli elementi della stessa riga per spazi oppure virgole e separando le diverse righe con punti e virgola. Le entrate possono essere numeri reali oppure complessi

```
A=[1 -5+3*i 7;9 0 4*i, -3 1 -1] %matrice 3x3
a=[1 3 5 7] %vettore riga (matrice 1x4)
b=[-9*i; 0] %vettore colonna (matrice 2x1)
k=-0.05 %scalare
```

Supponiamo che ci venga data una matrice $A = (a_{ij})$ con $i = 1 \dots m$, $j = 1 \dots n$. Osservando che in MATLAB gli indici partono sempre da 1, i seguenti comandi possono essere utili per accedere agli elementi di una matrice oppure un vettore dato.

`A(i,j)`	elemento a_{ij} della matrice A
`A(i,:)`	riga i-esima della matrice A
`A(:,j)`	colonna j-esima della matrice A
`A(i1:i2,j1:j2)`	sottomatrice di A per cui $i_1 \leq i \leq i_2$, $j_1 \leq j \leq j_2$
`a(i)`	elemento i-esimo del vettore a

Richiamiamo brevemente le operazioni principali applicabili alle matrici, facendo attenzione alle usuali **regole di compatibilità** delle operazioni matriciali

+	`A+B` somma
-	`A-B` divisione
*	`A*B` moltiplicazione
^	`A^k` potenza
/	`B/A=B*`A^{-1} divisione a destra
\	`A\B=`A^{-1}`*B` divisione a sinistra
'	`A'` trasposta di A

Nel caso in cui si vogliono effettuare delle operazioni **puntuali**, diverstamente chiamate anche di **array**, che operano *elemento per elemento*, gli operatori elencati sopra devono essere utilizzati con un punto davanti : `.* ./ .^ .\`. Altre funzioni importanti che si utilizzano spesso per vettori sono `max, min, median, sort, sum, prod, length`. Per più informazioni sulla loro sintassi si faccia riferimento al comando `help` di MATLAB.

In seguito richiamiamo *funzioni predefinite* in MATLAB per la creazione di matrici particolari

`eye(n)`	matrice identità $n \times n$
`zeron(m,n)`	matrice $m \times n$ di soli zeri
`ones(m,n)`	matrice $m \times n$ di soli uni
`rand(m,n)`	matrice $m \times n$ ad entrate casuali di valori tra 0 e 1
`diag(a)`	matrice quadrata diagonale di dimensione $n \times n$ con gli elementi di a sulla diagonale

Infine, ricordiamo che data una matrice quadrata A di dimensione n, il comando `diag(A)` restituisce una un vettore colonna di n elementi pari a quelli sulla diagonale di A.

Altre funzioni importanti operanti su matrici sono

`inv`	`inv(A)` inversa di una matrice quadrata A
`size`	`size(A)` dimensioni di una matrice A
`det`	`det(A)` determinante di una matrice quadrata A
`rank`	`rank(A)` rango di una matrice A
`eig`	`y=eig(A)` vettore colonna con gli autovalori di A

È consigliabile l'utilizzo del comando `cond`, prima di richiamare il comando `det`, per determinare il numero di condizionamento di una data matrice quadrata, e dunque verificare sesia invertibile o meno. Se vogliamo visualizzare anche gli autovettori di una matrice quadrata A, possiamo fare riferimento al comando `[U,D]=eig(A)`, che restituisce una matrice U le cui colonna sono gli autovettori di A, ed una matrice diagonale D i cui elementi sulal diagonale sono gli autovalori di A.

Ultima versione aggiornata: 22 Aprile 2020

Referenze

1. MATLAB® The language of technical computing: computation, visualization, programming, The MathWorks Inc
2. MATLAB: An introduction with applications, A. Gilat, Wiley
3. Scientific Computing with MATLAB and Octave, A. Quarteroni, , F. Saleri, P. Gervasio, Springer
4. MATLAB Guide1 D. J. Higham and N. J. Higham, SIAM
5. Numerical Computing with MATLAB, C. Moler, SIAM