

Introduzione a Matlab (*e al Calcolo Numerico*)

Giuseppe Rodriguez
Università di Roma "Tor Vergata"

Seminario nell'ambito del corso di
Fondamenti di Informatica
per gli studenti di Ingegneria Meccanica e Ingegneria Chimica
Università di Cagliari - A.A. 2000/2001

Matlab

- Ambiente integrato per il calcolo scientifico e la visualizzazione scientifica
- Interfaccia verso librerie scientifiche di pubblico dominio (Lapack, FFTW, etc.)
- Tutte le subroutines di calcolo sono documentate
- Interprete + Linguaggio di programmazione
- Utilizzato per didattica, ricerca e sviluppo

⇒ dispense e software sono disponibili

Operazioni su scalari 1

- Variabili e controllo del *workspace* (`whos`, `clear`, `save` e `load`)
- Costanti, notazione esponenziale, controllo dell'output (`format` e `;`)
- Operazioni aritmetiche: `+`, `-`, `*` e `/`. Parentesi.
- Potenze (`^`), radici (`sqrt`), funzioni trascendenti, `pi` greco
- Editing della linea di comando
- Maiuscole e minuscole
- Documentazione in linea: `help`, `doc` e `demo`

Operazioni su scalari 2

- Numeri complessi
 - `i`, `abs` e `angle`
 - Esempio: risoluzione di una equazione di secondo grado
- Operazioni “proibite”: `inf` e `Nan`

Analisi Numerica

L'Analisi Numerica è lo studio degli algoritmi per i problemi della matematica del continuo.

- Caratteristiche di un algoritmo (es.: equazioni di secondo grado):
 - **stabilità**
 - **complessità computazionale**
 - **occupazione di memoria**

Analisi Numerica

L'Analisi Numerica è lo studio degli algoritmi per i problemi della matematica del continuo.

- Caratteristiche di un algoritmo (es.: *equazioni di secondo grado*):
 - **stabilità**
 - **complessità computazionale**
 - **occupazione di memoria**
- Ulteriori insidie possono essere nascoste nel problema da risolvere:
 - buona o cattiva **posizione**
 - **corrispondenza** col problema fisico reale
 - **condizionamento**

Analisi Numerica

- Caratteristiche di un algoritmo:
 - **stabilità** (propagazione degli errori)
 - **complessità computazionale** (numero di operazioni richieste, è proporzionale al tempo di calcolo)
 - **occupazione di memoria**
- Ulteriori insidie possono essere nascoste nel problema da risolvere:
 - buona o cattiva **posizione** (esiste una e una sola soluzione?)
 - **corrispondenza** col problema fisico reale (i risultati ottenuti sono significativi?)
 - **condizionamento** (amplificazione degli errori sui dati, a prescindere dall'algoritmo utilizzato)

Sorgenti di errore

- Dati sperimentali
- Semplificazioni introdotte nel modello matematico
- **Aritmetica di macchina** (arrotondamento e/o calcoli precedenti)

Conseguenza 1: Tutti i numeri sono approssimati

Conseguenza 2: I computer “sbagliano” (commettono errori) sistematicamente

Esempi:

$$3 \left(\frac{4}{3} - 1 \right) - 1$$

$$1 + \frac{\text{eps}}{2}$$

$$\frac{1 - \cos x}{x^2} \quad \text{vs.} \quad \frac{1}{2} \left(\frac{\sin \frac{x}{2}}{\frac{x}{2}} \right)^2 \quad \text{per } x \rightarrow 0$$

Vettori e matrici

- Definizione estensiva di vettori e matrici
- Definizione intensiva di vettori: l'operatore *colon* (:)
- Controllo del *workspace* (*whos*, *clear*, *save* e *load*)
- *Patchwork* (concatenazione) di arrays. Il vettore nullo ([])
- Accesso a singoli elementi e a sottoarrays (e loro modifica)
- Funzioni che generano arrays: *ones*, *zeros*, *magic*, *rand*, *randn*, *eye*, *diag*
- Funzioni statistiche: *sum*, *mean*, *std*, *min*, *max*

Algebra lineare 1

- Tutte le operazioni sono matriciali (a parte la divisione)
- Trasposizione (`'`), determinante (`det`) e inversa (`inv`, ma non solo)
- Norme di vettori (`norm`)
- Esempi:
 - Prodotti di arrays: `5*x`, `5*A`, `A*x`, `x'*A`, `A*B`, `x'*y`
 - Verificare che `magic(7)` sia un quadrato magico
 - Valutazione della complessità computazionale (`flops`)
 - Valutazione del tempo di calcolo (`tic` e `toc`)
 - **Complessità computazionale** del calcolo del determinante con la regola di Laplace e con l'algoritmo di Gauss (`det`)

Risoluzione di un sistema di n equazioni lineari

Tempi di calcolo su un calcolatore che effettua un milione di operazioni al secondo (1 Mflop/s)

n	Cramer + Laplace	Gauss
12	103 minuti	$7.1 \cdot 10^{-4}$ secondi
13	24 ore	$8.9 \cdot 10^{-4}$ secondi
14	15 giorni	$1.1 \cdot 10^{-3}$ secondi
20	1.620.083 anni	$3.1 \cdot 10^{-3}$ secondi
30	$2.6 \cdot 10^{20}$ anni	$9.9 \cdot 10^{-3}$ secondi
40	$1.1 \cdot 10^{36}$ anni	$2.3 \cdot 10^{-2}$ secondi
50	$4.9 \cdot 10^{52}$ anni	$4.4 \cdot 10^{-2}$ secondi

Risoluzione di un sistema di n equazioni lineari

Tempi di calcolo su un calcolatore che effettua un milione di operazioni al secondo (1 Mflop/s)

n	Cramer + Laplace	Gauss
12	103 minuti	$7.1 \cdot 10^{-4}$ secondi
13	24 ore	$8.9 \cdot 10^{-4}$ secondi
14	15 giorni	$1.1 \cdot 10^{-3}$ secondi
20	1.620.083 anni	$3.1 \cdot 10^{-3}$ secondi
30	$2.6 \cdot 10^{20}$ anni	$9.9 \cdot 10^{-3}$ secondi
40	$1.1 \cdot 10^{36}$ anni	$2.3 \cdot 10^{-2}$ secondi
50	$4.9 \cdot 10^{52}$ anni	$4.4 \cdot 10^{-2}$ secondi

Alcune date:

inizio del giurassico: $1.9 \cdot 10^6$ anni
età della Terra: $4.6 \cdot 10^9$ anni
età dell'Universo: 10^{10} anni

Operazioni su arrays (*dot operations*)

- Operatori che agiscono componente per componente: +, -, .*, ./ e .^
- Applicazione: **grafico di una funzione**
 1. campionamento della variabile indipendente
 2. calcolo della funzione sui punti di campionamento
 3. tracciamento del grafico

```
x = [-5:.1:5]';  
y = x.^2;  
plot(x,y)
```

Esempi:

$$\sin(\pi x) + \frac{1}{5} \cos(7\pi x), \quad x \in [-1, 1]$$

$$\frac{1}{1 + 25x^2}, \quad x \in [-1, 1]$$

$$\frac{\sin(10x)}{10x}, \quad x \in [-1, 1]$$

Istruzioni grafiche

- Visualizzazione contemporanea di più serie di dati
- Apertura di più finestre grafiche
- Modifica dello stile delle linee
- Annotazioni: `title`, `legend`, `xlabel` e `ylabel`
- Altri tipi di grafici: `bar`, `stairs`, `stem`, `pie`, `hist`

Algebra lineare 2

- Risoluzione di sistemi lineari: $x=A\backslash b$;
- Generazione di un sistema lineare *test*
- **Cattivo condizionamento**: la matrice di Hilbert
- Calcolo di autovalori e autovettori: `eig` e `poly`

Altri argomenti

- Calcoli con polinomi:
 - valutazione: `polyval`
 - calcolo delle radici: `roots`, `poly`
 - derivazione e integrazione: `polyder`, `polyint`
 - prodotto e divisione: `conv`, `deconv`
- Approssimazione di funzioni
 - interpolazione: `interp1`
 - approssimazione ai minimi quadrati: `polyfit`
- Integrazione numerica: `quad`