Introduzione a Matlab

Giuseppe Rodriguez

Dipartimento di Matematica e Informatica

Università di Cagliari

12 e 13 febbraio 2024

Matlab

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

Materiale didattico:

- Il programma contiene un manuale in linea molto completo
- Su Internet si trovano manuali e dispense (Matlab Primer)
- Esistono cloni Open Source: Octave

Operazioni su scalari

- Costanti, variabili, maiuscole e minuscole.
- Prendere appunti con la funzione `diary`.
- Operazioni aritmetiche: `+`, `-`, `*` e `/`. Parentesi.
- Potenze ($\wedge$), radici (`sqrt`), funzioni trascendenti.
- Editing della linea di comando.
- Documentazione in linea: `help`, `doc` e `demo`.
- Valutazione del tempo di calcolo: `tic` e `toc` (funzione).
- Operazioni illegali: `Inf` e `NaN`.

Controllo della memoria e dell'output.

- Variabili e controllo del *workspace*: `who`, `whos`, `clear`.
- Le funzioni `single` e `double`.
- Notazione esponenziale, controllo dell'output: `format` e `;`.
- Memorizzare variabili: `save` e `load`.
- Variabili predichiarate: `pi` greco, `eps`, `i` e `1i`.
- Numeri complessi e funzioni per il loro uso: `real`, `imag`, `abs`, `angle`, `conj`.
- Approssimazione e arrotondamento: `ceil`, `floor`, `fix`, `round`, `sign`.
- Altri tipi di variabili: `struct`, `cell`, `table`.

Sorgenti di errore

I numeri su cui si opera sono tipicamente affetti da errori

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

Conseguenze

1. Tutti i numeri sono approssimati
2. I computer "sbagliano" (commettono errori nei calcoli) sistematicamente

Esempi sui numeri di macchina e la propagazione degli errori

- `3*0.1-0.3` e `3*single(0.1)-0.3`
- `sqrt(2)^2-2`
- $3\left(\frac{4}{3}-1\right)-1$
- $q = 1 + \frac{\text{eps}}{2}$, $q - 1$
- $v = \begin{bmatrix} 10^{\pm 200} \\ -10^{\pm 200} \end{bmatrix}$, $\sqrt{v_1^2 + v_2^2}$ vs. `norm(v)`
- $\frac{1-\cos x}{x^2}$ vs. $\frac{1}{2}\left(\frac{\sin\frac{x}{2}}{\frac{x}{2}}\right)^2$ per $x \to 0$, $(x = 10^{-1}, 10^{-2}, \ldots)$

Vettori e matrici

- Definizione estensiva di vettori e matrici.
- Definizione intensiva di vettori: l'operatore *colon* (`:`).
- *Patchwork* (concatenazione) di arrays. Il vettore nullo (`[]`).
- Controllo delle dimensioni: `size`, `length`, `whos`.
- Funzioni che generano arrays: `ones`, `zeros`, `magic`, `rand`, `randn`, `eye`, `diag`, `toeplitz`.
- Funzioni per manipolare arrays: `rot90`, `tril`, `triu`.
- Accesso e modifica di singoli elementi e sottoarrays. Subindexing.
- Funzioni statistiche: `sum`, `mean`, `std`, `min`, `max`. Loro applicazione a vettori e matrici.
- Ottimizzazione del codice: preallocazione di un array.

Programmazione

- Creazione di uno *script*. Esempio: costruzione di una matrice mosaico.
- Commenti in un programma, help in linea
- Trasformazione dello script in una function. Dichiarazione e parametri.
- Funzioni vs. scripts: I/O, variabili automatiche.
- Controllo del numero dei parametri: `nargin` e `nargout`. Assegnazione di valori di default.
- Controllo degli errori: `warning` e `error`.
- Controllo dell'iterazione: cicli `for`.
- Meglio functions come files esterni, o dentro scripts e funzioni?

Calcolo vettoriale e matriciale

- Tutte le operazioni sono matriciali (a parte la divisione).
- Esempi:
  - Prodotti di arrays: `5*x, 5*A, A*x, x'*A, A*B, x'*y, x*y'`
  - `Verificare che magic(7) sia un quadrato magico`
- Trasposizione (`'`), `determinante (det), norma (norm), numero di condizionamento (cond).`
- `Inversa (inv, ma non solo), autovalori e autovettori (eig).`
- `Risoluzione di sistemi lineari.`
- `Significato dell'operatore di divisione matriciale (\ e /).`
- `Esempio: risoluzione di una successioni di sistemi lineari di dimensione crescente con matrice mosaico, controllo degli errori e loro visualizzazione (meglio plot o semilogy?).`

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 di base

- 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`.

- Le istruzioni `input` e `fprintf`.
- Operatori e variabili logiche.
- Controllo dell'iterazione: cicli `while`.
- Istruzioni condizionali: `if` e `switch`.
- Esempi
  1. Equazioni di secondo grado
  2. Metodo di Newton
  3. Calcolo di e^x (e di $n!$)
  4. Sistemi triangolari

Subindexing

- Selezione di componenti di un vettore mediante *subindexing*.
- Indicizzazione con vettori di indici e di 0-1.
- Selezione di elementi con condizioni logiche e `find`.
- `Vantaggi nella programmazione.`
- `Esempi`
  1. `...`

Funzioni di libreria

- 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, integral`
- `Risoluzione di equazioni: fzero, roots`
- `Risoluzione di ODE: ode23, ode45`
- `Calcolo simbolico: sym, syms, int, semplificazione di formule.`