

Introduzione a Matlab

Giuseppe Rodriguez

Dipartimento di Matematica e Informatica

Università di Cagliari

Laboratorio di Calcolo Numerico
Laurea Magistrale in Ingegneria per l'Ambiente e il Territorio

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

- 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)

- 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
- Esistono cloni Open Source: Octave

- Costanti, variabili, maiuscole e minuscole
- Operazioni aritmetiche: +, -, * e /. Parentesi.
- Prendere appunti con la funzione `diary`
- Potenze (\wedge), radici (`sqrt`), funzioni trascendenti
- Editing della linea di comando
- Valutazione del tempo di calcolo (`tic` e `toc`)
- Notazione esponenziale, controllo dell'output (`format` e `;`)
- Variabili e controllo del *workspace* (`whos`, `clear`, `save/load`)
- Numeri complessi e funzioni per il loro uso
- Variabili predichiarate: `pi` greco, `i`, `eps`
- Documentazione in linea: `help`, `doc` e `demo`

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

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

- Definizione estensiva di vettori e matrici
- Definizione intensiva di vettori: l'operatore *colon* (:)
- *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
- Ottimizzazione del codice: preallocazione di un array
- Altri tipi di variabili: struct e cell

- Tutte le operazioni sono matriciali (a parte la divisione)
- Trasposizione (') e determinante (det)
- inversa (inv, ma non solo)
- Norme di vettori (norm)
- Esempi:
 - Prodotti di arrays: $5*x$, $5*A$, $A*x$, $x'*A$, $A*B$, $x'*y$, $x*y'$
 - Verificare che `magic(7)` sia un quadrato magico

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)
```

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

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

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

- 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`

- Risoluzione di sistemi lineari: $x=A\backslash b$;
- Calcolo di autovalori e autovettori: `eig` e `poly`
- Generazione di un sistema lineare `test`
- **Cattivo condizionamento**: la matrice di Hilbert
- Esempi:
 - Sperimentazione sulla risoluzione di sistemi lineari, con visualizzazione di errori, condizionamento, tempi di calcolo

- Creazione di uno *script*
- Le istruzioni `input` e `fprintf`
- Istruzioni condizionali: `if` e `switch`
- Operatori e variabili logiche
- Controllo dell'iterazione: `while` e `for`
- Commenti in un programma, `help` in linea
- Funzioni vs. scripts: `I/O`, `nargin` e `nargout`.
- Esempi
 - 1 Equazioni di secondo grado
 - 2 Metodo di Newton
 - 3 Calcolo di e^x (e di $n!$)
 - 4 Sistemi triangolari

- 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 ...

- 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`