Laboratorio di Informatica T (Ch5)

Grafici Cartesiani con Matlab

Grafici Con Matlab

Matlab permette di disegnare facilmente dei grafici

La prima cosa da fare è costruire una nuova "figura":

```
figure()
```

- La funzione **`figure`** apre una nuova finestra…
- …In cui verrà inserito il disegno

In molti casi, questo passaggio può essere saltato

- Se non è stata ancora costruita una figura…
- …Molte funzioni di disegno se ne accorgono…
- …E chiamano **`figure`** automaticamente

Grafici Con Matlab

Supponiamo di volere costruire un grafico cartesiano

Per prima cosa dobbiamo costruire il vettore delle x

Possiamo usare `linspace`:

```
x = linspace(-2*pi, 2*pi, 200)
```

- Provandolo, noterete che il vettore restituito contiene molti elementi
- Potete disabilitare la visualizzazione aggiungendo “;”

Quindi scrivendo:

```
x = linspace(-2*pi, 2*pi, 200);
```

…Si ottiene lo stesso risultato, ma senza stampe

Plotting

Ora otteniamo il vettore con i valori per l'asse delle $\mathbf{y}$

Calcoliamo per esempio la funzione "seno"

- Basta applicarla al vettore **`x`**...
- ...Perché **`sin`** opera elemento per elemento

```
y = sin(x);
```

In questo modo otteniamo:

- I valori della funzione **`sin`**...
- ...Corrispondenti agli elementi del vettore **`x`**

Plotting

A questo punto possiamo disegnare il grafico

Si utilizza la funzione `plot`

```
plot(x, y)
```

- `plot` riceve come argomento due vettori (coordinate x e y)
- Il disegno viene ottenuto “congiungendo i puntini”

Avremmo anche potuto scrivere direttamente:

```
plot(x, sin(x)) % senza usare una variabile per y
```

Si può anche specificare un colore:

```
plot(x, sin(x), 'b') % b = blue
```

- Si può specificare lo stile della linea, lo spessore…
- Guardate la documentazione di `plot` per altri dettagli!

Plotting

Si può aggiungere una griglia con:

```
grid()
```

Per disegnare più curve sovrapposte:

```
figure()                    % Nuova figura
plot(x, sin(x), 'b')        % Prima curva, in blu
hold on                     % Attiva la modalità "hold"
plot(x, cos(x), 'g')        % Seconda curva, in verde
hold off                    % Disattiva la modalità "hold"
```

- Senza la modalità **`hold`** ogni plot rimpiazza il precedente

Esercizio: Grafici Cartesiani

Esercizio: Grafici Cartesiani

Disegnate, per $x \in [-4, 4]$ le seguenti funzioni:

$$(1) \quad x^2 - x \qquad\qquad (2) \quad \frac{1}{1+|x|}$$

$$(3) \quad \frac{1}{1+e^{-x}} \qquad\qquad (4) \quad \frac{1}{x|x|}$$

Prima di disegnarle, cercate di intuire se sono:

- Continue
- Derivabili (senza “spigoli”)
- Concave/convesse (eventualmente)

E quindi verificatelo visivamente

Grafici Cartesiani (Soluzione)

Soluzione:

```
x = linspace(-4, 4, 200);   % Notate il ;
plot(x, x.^2 - x)           % Cont., derivabile, conv.
plot(x, 1./(1 + abs(x)))    % Cont., non derivabile
plot(x, 1./(1 + exp(-x)))   % Cont., derivabile
plot(x, 1./(x.*abs(x)))     % Non cont., non derivabile
```

- Notate che per l'ultima funzione il grafico ha un artefatto!
- La funzione non è continua, ma si vede comunque una linea
- Succede perché in `0` la funzione non è definita…
- …ma il punto subito prima e subito dopo sono ben definiti…
- …quindi `plot` li congiunge con una linea

Esempio: Serie Armonica

Esempio: Serie Armonica

Consideriamo la serie armonica (troncata):

$$S_N = \sum_{n=1}^{N} \frac{1}{n}$$

Vogliamo verificare empiricamente che S_n diverge per $N \to \infty$

- Calcoliamo il valore di S_N con $N = 1..N_{max}$
- Riportiamo i valori di N ed S_N su un grafico

Esempio: Serie Armonica

Consideriamo la serie armonica (troncata):

$$S_N = \sum_{n=1}^{N} \frac{1}{n}$$

Vogliamo verificare empiricamente che S_n diverge per $N \to \infty$

- Calcoliamo il valore di S_N con $N = 1..N_{max}$
- Riportiamo i valori di N ed S_N su un grafico

Osservazione fondamentale:

$$S_N = S_{N-1} + \frac{1}{N}$$

- Quindi possiamo iterate sui valori di N…
- …E ad ogni passo salvare S_N in un vettore
- Al passo successivo, useremo S_N per calcolare S_{N+1}

Esempio: Serie Armonica

Una possibile implementazione (nello start-kit):

```
clear all

Nmax = 50;
y = [];       % Preparo un vettore per i risultati

s = 0;
for n = 1:Nmax
    s = s + 1/n;
    y(n) = s;
end

plot(1:Nmax, y)
```

Esempio: Risultati su Vettori/Matrici

Esempio: Risultati su Vettori/Matrici

Dato un vettore di valori $\mathbf{x}$:

- Vogliamo calcolare i valori di $\sin(x)$ e $\cos(x)$
- I valori di $\sin(x)$ dovranno comparire come prima colonna del risultato y
- I valori di $\cos(x)$ dovranno comparire come seconda colonna del risultato y

Come procedere:

- Calcolare $\sin$ e $\cos$ è banale (ci sono le funzioni predefinite)
- La cosa complicata è come salvare i risultati su y
- Nota che y dovrà essere una matrice

Vediamo tre metodi (nel file `my_sin_cos.m`)...

Esempio: Risultati su Vettori/Matrici

Primo metodo: concatenazione di colonne

```
x = linspace(-pi, pi);

y = [sin(x).', cos(x).'];
```

- **`sin(x)`** e **`cos(x)`** restituiscono un vettore riga
- Quindi vanno trasposti per poter essere correttamente concatenati

Avremmo potuto anche usare:

```
y = [sin(x); cos(x)]     % Due righe una sopra l'altra
y = y.'                  % Traspongo y
```

Esempio: Risultati su Vettori/Matrici

Secondo metodo: concatenazione di righe (una per una)

```
y2 = [];
for i = 1:length(x)
        row = [sin(x(i)), cos(x(i))];
    y2 = [y2; row];
end
```

- Ad ogni passo estendiamo costruiamo una riga con $\sin(x_i)$ e $\cos(x_i)$
- Quindi estendiamo **`y2`** aggiungendo una riga

Esempio: Risultati su Vettori/Matrici

Terzo metodo: assegnamento di righe (una per una)

```
y3 = [];
for i = 1:length(x)
    y3(i, :) = [sin(x(i)), cos(x(i))];
end
```

- Ad ogni passo estendiamo costruiamo una riga con $\sin(x_i)$ e $\cos(x_i)$
- Quindi assegniamo la riga nella posizione corretta
- Per effettuare l'assegnamento occorre usare due indici (i.e. `i` e `:`)
- In questo modo Matlab capisce che vogliamo assegnare una intera riga

Esercizio: Serie Geometrica

Esercizio: Serie Geometrica

Consideriamo la serie geometrica (troncata):

$$S_N = \sum_{n=0}^{N} ar^n$$

Vogliamo verificare empiricamente che S_n converge se $r < 1$, per $N \to \infty$

- Calcoliamo il valore di S_N con $N = 0..N_{max}$
- Riportiamo i valori di N ed S_N su un grafico

Osservazione fondamentale:

$$S_N = S_{N-1} + ar^N$$

- Quindi possiamo iterate sui valori di $N\ldots$
- …E ad ogni passo salvare S_N in un vettore
- Al passo successivo, useremo S_N per calcolare S_{N+1}

Svolgete l'esercizio nel file di script `my_geometic.m`

Esercizio: Somma di Matrici

Esercizio: Somma di Matrici

Nel file di script `my_msum`:

- Sono definite due matrici **`A`** e **`B`** di esempio, con la stessa dimensione
- Calcolatene la somma

Confrontate il vostro risultato con quello dell'operatore **`+`** di Matlab

Esercizio: Vettore di Indici

Esercizio: Indicizzazione con Vettore di Indici

Abbiamo visto che Matlab permette di accedere ad un vettore con:

```
V(I)
```

- **V** è un vettore
- **I** è un vettore di indici

Nel file di script `my_index.m`:

- Sono inizializzati un vettore **X** di dati ed un vettore **I** di indici
- Calcolate il vettore **Y** con gli elementi di **X**…
- …Alle posizioni specificate da **I**

Confrontate il vostro risultato con quello di Matlab

Esercizio: Funzione Esponenziale

Esercizio: Funzione Esponenziale

Il valore di $\mathbf{e^x}$ si può calcolare sfruttando la serie di Taylor:

$$S_N = \sum_{k=0}^{N} \frac{x^k}{k!}$$

Vogliamo verificarlo empiricamente che $S_N \to e^x$ quando $N \to \infty$:

- Calcoliamo il valore di S_N con $N = 0..N_{max}$
- Riportiamo i valori di N ed S_N su un grafico
- Confrontiamo il valore di $S_{N_{max}}$ e e^x
- Per calcolare il fattoriale: `factorial(k)`

Come nel caso della serie armonica, vale:

$$S_N = S_{N-1} + \frac{x^N}{N!}$$

Svolgete l'esercizio nel file di script `my_exponential.m`

Esercizio: Funzione Seno

Esercizio: Funzione Seno

Il valore di $\sin(x)$ si può calcolare sfruttando la serie di Taylor:

$$S_N = \sum_{k=0}^{N} \frac{-1^k x^{2k+1}}{(2k+1)!}$$

- Vale che $S_N \rightarrow \sin(x)$ quando $N \rightarrow \infty$

Vogliamo verificarlo empiricamente:

- Calcoliamo il valore di S_N con $N = 0..N_{max}$
- Riportiamo i valori di N ed S_N su un grafico
- Confrontiamo il valore di $S_{N_{max}}$ e $\sin(x)$

Come nel caso della serie armonica, vale:

$$S_N = S_{N-1} + \frac{-1^N x^{2N+1}}{(2N+1)!}$$

Svolgete l'esercizio nel file di script `my_sin.m`

Esercizio: Prodotto Cartesiano

Esercizio: Prodotto Cartesiano

Dati due vettori $\mathbf{x}$ ed $\mathbf{y}$ di $\mathbf{n}$ elementi, il loro prodotto cartesiano è:

$$\begin{pmatrix} x_1 & y_1 \\ x_1 & y_2 \\ \vdots & \vdots \\ x_n & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix}$$

Nel file di script **`my_cartesian.m`**:

- Si calcoli il prodotto cartesiano dei due vettori di test
- L'aspetto più complesso è indicizzare il vettore dei risultati
  - Un metodo: concatenare nuove righe!
  - Un altro metodo: assegnare nuove righe