

1. Dato il file `times_series.mat` si scriva una funzione matlab che implementi un filtro a media mobile. Tale funzione deve avere come **parametri di input**: una stringa `file_name` (che contiene il nome del file da importare), un intero `N` (che indica l'ampiezza della finestra del filtro a media mobile), un intero `M` (che indica su quanti campioni del segnale vogliamo applicare il filtro a media mobile). La funzione deve restituire come **output** il segnale filtrato `Yf`.

In particolare la funzione deve

- 1) importare il file `file_name` del segnale, e memorizzare il suo contenuto in un vettore `Y`
- 2) determinare la lunghezza del segnale `Y` e stampare a video il suo valore
- 3) selezionare i primi `M` campioni del segnale e assegnarli alla nuova variabile `Y2`
- 4) disegnare in un grafico il segnale `Y2`
- 5) implementare un **filtro a media mobile**: si fa scorrere una finestra di ampiezza molto piccola `N` (e solitamente dispari) sul segnale `Y2` e si sostituisce al valore del campione corrispondente al centro della finestra il valore che deriva dalla media dei campioni compresi nella finestra. Il risultato di questa operazione di filtraggio sarà un nuovo segnale `Yf`
- 6) disegnare in una stessa finestra grafica il segnale originale `Y2` e la sua versione filtrata `Yf`
- 7) salvare il grafico ottenuto al punto 6) in un file.

Scrivere una relazione in cui si spieghi l'algoritmo implementato e in cui si riportino i grafici del segnale originale `Y2` e di quello filtrato `Yf` per diversi valori di `N` (ad es. `N=3,5,15,...`) e di `M` (ad es `M=100,1000`). Includere una breve descrizione del filtro a media mobile.

2. Dato il file meg_data.mat si scriva una funzione matlab che implementi l'operazione di media su un segnale continuo a partire da un file di trigger.

Tale funzione deve avere come **parametri di input**: una stringa data_filename (che contiene il nome del file dei dati da importare), una stringa trigger_filename (che contiene il nome del file dei trigger da importare). La funzione deve restituire come **output** la nuova matrice dei dati B. In particolare la funzione deve

- 1) importare i file data_filename e trigger_filename e memorizzare il loro contenuto nelle due variabili: Braw e trigger. Braw è la matrice dei dati: ogni riga i contiene il segnale continuo registrato dall'i-esimo sensore (le colonne rappresentano gli istanti temporali in cui è stato registrato il segnale); trigger è un vettore e ogni suo elemento indica l'istante in cui si è verificato un certo evento (ad es. la comparsa di un'immagine)
- 2) determinare le dimensioni della matrice dei dati Braw e del vettore dei trigger e stamparle a video
- 3) disegnare la matrice dei dati Braw rispetto al tempo
- 4) costruire a partire da Braw una nuova matrice di dati B ottenuta mediando il segnale di ogni sensore in base ai trigger: per ogni segnale registrato da un sensore (ogni riga di Braw) si prendono N (numero di trigger) sottosegnali (che si chiamano epoche) ognuno costruito prendendo 50 campioni prima di trigger(i) e 300 campioni dopo trigger(i). Mediare le N epoche così ottenute. Questa operazione di selezione delle epoche e media va ripetuta per ogni sensore (ossia per ogni riga della matrice Braw).
- 5) disegnare in una stessa finestra grafica la nuova matrice dei dati B così ottenuta e la matrice dei dati contenente tre epoche
- 6) salvare il grafico ottenuto al punto 5) in un file.

Scrivere una breve relazione in cui si spieghi l'algoritmo implementato e in cui si riportino i grafici della matrice dei dati mediati B e di tre epoche differenti della matrice dei dati continui Braw. Includere una breve descrizione sulla MagnetoEncefalografia (la tecnica con la quale sono stati registrati questi dati).

3. Dato il file cameraman_noise.tif si scriva una funzione matlab che implementi un filtro mediano sull'immagine importata. Tale funzione deve avere come **parametri di input**: una stringa file_name (che contiene il nome dell'immagine da importare) e un intero N (che indica l'ampiezza della finestra del filtro mediano). La funzione deve restituire come **output** la nuova matrice filtrata If.

La funzione deve

- 1) importare il file file_name dell'immagine, e memorizzare il suo contenuto in una matrice I
- 2) determinare le dimensioni dell'immagine I importata
- 3) disegnare l'immagine usando le apposite funzioni di Matlab
- 4) implementare un filtro mediano: è un filtro digitale che viene utilizzato per eliminare il rumore da immagini digitali. Ogni valore di grigio di un pixel è sostituito dalla mediana (**non la media**) dei valori contenuti in una finestra centrata nel pixel. Il risultato di questa operazione di filtraggio sarà la nuova immagine filtrata If
- 5) disegnare in una stessa finestra grafica l'immagine originale I e la sua versione filtrata If
- 6) salvare il grafico ottenuto al punto 5) in un file.

Scrivere una breve relazione in cui si spieghi l'algoritmo implementato e in cui si riportino i grafici dell'immagine originale I e della sua versione filtrata If per diversi valori di N (ad es. N=9,25,36,...). Includere una breve descrizione del filtro mediano.

4. L'oscillatore di van der Pol è un oscillatore con uno smorzamento non lineare descritto dall'equazione differenziale del secondo ordine

$$y'' - \varepsilon (1 - y^2) y' + y = 0,$$

dove $\varepsilon > 0$ è un parametro che determina l'ampiezza dello smorzamento.

Il modello è stato proposto nel 1920 da Balthasar van der Pol (1889-1959), ingegnere della Philips Company (Olanda), per lo studio dei triodi.

Dopo aver trasformato l'equazione del secondo ordine in un sistema di due equazioni del primo ordine, scrivere una funzione matlab che abbia come **parametri di input** ε e i valori iniziali $y(0)$ e $y'(0)$ e restituisca come **output** una approssimazione della soluzione dell'equazione di van der Pol.

In particolare la funzione deve

- 1) usare un metodo del IV ordine per approssimare la soluzione dell'equazione di van der Pool fino al terzo zero di $y(t)$
- 2) approssimare i primi tre zeri di $y(t)$ con una accuratezza minore di 10^{-6}
- 3) controllare che siano dati in input tutti i parametri e restituire un messaggio di errore in caso contrario
- 4) essere richiamata con $\varepsilon=0.1$ e valori iniziali $y(0)=1$ e $y'(0) = 0$

Scrivere una breve relazione sul lavoro svolto includendo una breve ricerca sulla storia dell'equazione di van der Pol. Vedere cosa accade per altri valori interessanti dei parametri.

Opzionale: studiare cosa succede al variare di ε e studiare l'esistenza di soluzioni periodiche.

5. Il modello di Fitzhugh-Nagumo

$$\begin{aligned}V' &= c(V - V^3/3 + W + I) \\W' &= -\frac{1}{c}(V - a + bW)\end{aligned}$$

è una versione semplificata del modello di Hodgkin-Huxley che descrive la generazione di spike degli assoni di calamaro gigante. V è il potenziale della membrana, W è un debole feedback negativo e I è l'ampiezza della corrente di stimolo.

Scrivere una funzione matlab che abbia come **parametri di input** a, b, c, I , e i valori iniziali $V(0)$ e $W(0)$ e restituisca come **output** l'approssimazione della soluzione del modello.

La funzione deve

- 1) usare un metodo del IV ordine per approssimare la soluzione del modello di FitzHugh-Nagumo
- 2) controllare che siano dati in input tutti i parametri richiesti e restituire un messaggio di errore in caso contrario
- 3) essere richiamata con $a=0.7$ $b=0.8$ $c=3$ $I=0$ e condizioni iniziali $V(0)=1$ e $W(0)=0$
- 4) creare il grafico delle soluzioni nello spazio delle fasi

Scrivere una breve relazione sul lavoro svolto includendo una breve ricerca storica sul modello di Fitzhugh-Nagumo. Individuare se esiste un punto di equilibrio. Vedere cosa accade per $I=-0.2$.

6. Nella tabella seguente sono riportate le misure di intensità luminosa di un'immagine aerea in funzione della posizione dei pixel

Posizione del pixel	-126	-84	-42	0	42	84	126
Intensità (DN)	119	165	231	243	244	214	136

Scrivere uno script matlab che

- 1) costruisca il polinomio algebrico $P(x)$ che meglio approssimi i dati nel senso dei minimi quadrati
- 2) utilizzi il polinomio individuato $P(x)$ per approssimare l'intensità luminosa nelle posizioni $i=128$, $i=-128$
- 3) utilizzi il polinomio $P(x)$ come modello per simulare nell'intervallo $[-128,128]$ un insieme di dati affetti da rumore gaussiano di ampiezza non superiore a 7 DN
- 4) plotti i dati simulati ottenuti al punto 3)

Scrivere una breve relazione in cui si spieghi l'algoritmo. Includere una breve descrizione del metodo dei minimi quadrati.

7. Un cane, libero in un prato, vede il padrone che cammina lungo la strada e corre verso di lui. Supponendo che il cane tenda sempre verso il padrone, e che la strada sia rettilinea, l'equazione della traiettoria del cane è

$$xy'' = c\sqrt{1 + (y')^2}$$

in cui c è il rapporto tra la velocità dell'uomo e quella del cane.

Dopo aver trasformato l'equazione del secondo ordine in un sistema di due equazioni del primo ordine, scrivere una funzione matlab che abbia come **parametri di input** c , i valori iniziali e il passo di integrazione, e restituisca come **output** una approssimazione della soluzione del sistema di equazioni.

In particolare la funzione deve

- 1) usare il metodo di Runge-Kutta del IV ordine per approssimare la soluzione del sistema di equazioni
- 2) controllare che siano dati in input tutti i parametri e restituire un messaggio di errore in caso contrario
- 3) disegnare il grafico della soluzione

Scrivere una breve relazione sul lavoro svolto. Vedere cosa accade per diversi valori del passo di integrazione. In particolare se $c = 1/2$, l'uomo parte da $(0,0)$ e il cane da $(1,0)$, in quale punto il cane raggiunge il padrone?

8. Scrivere una funzione MatLab che

- legga in input un vettore di dati (segnale discreto) Y e un numero intero $R \in [0 \ 100]$.
- calcoli la trasformata di Fourier discreta di Y usando la funzione MatLab **fft.m**.
- Selezioni lo R% dei coefficienti della trasformata privilegiando il primo coefficiente e quelli con valore assoluto maggiore, mentre si pongano a zero i rimanenti (nel calcolo del numero di coefficienti da preservare, si consideri il fatto che la trasformata è simmetrica e i coefficienti sono numeri complessi).
- Costruisca e restituisca in output un'approssimazione Yapp del segnale originale Y invertendo la trasformata (mediante la funzione MatLab **ifft.m**) usando i coefficienti modificati secondo quanto descritto al punto precedente.
- Calcoli e restituisca in output lo scarto quadratico medio Err calcolato tra Y e Yapp.

Si considerino i seguenti segnali

- segnale costante a tratti così definito
Y=zeros(1,100);
Y(N/4+1:3/4*N) = 1;
- segnale somma di tre coseni così definito
y1 = cos(2*pi*20*(0:999)/1000);
y2 = cos(2*pi*80*(0:999)/1000);
y3 = cos(2*pi*120*(0:999)/1000);
Y= y1 + y2 + y3;
- segnale periodico a tratti così definito
y1 = cos(2*pi*20*(0:1:300)/1000);
y2 = cos(2*pi*80*(301:1:700)/1000);
y3 = cos(2*pi*120*(701:1:999)/1000);
Y=[y1 y2 y3];

Scrivere una breve relazione che includa le approssimazioni ottenute per ognuno di essi in corrispondenza dei seguenti valori di R = 100, 90, 50, 25, 10, 1.

Commentare i risultati ottenuti e, per ogni segnale, graficare la curva che ha in ascissa R e in ordinata il corrispondente valore Err.

9. Scrivere una funzione MatLab che

- legga in input una matrice I di dimensione $m \times n$, una variabile di tipo stringa *method* e un numero intero N ; se N non viene dato in input, assegni il valore 2
- restituisca in output la matrice J di dimensione $Nn \times Nm$ ottenuta interpolando opportunamente I usando il metodo indicato dalla variabile *method*. Per l'interpolazione si usino le funzioni predefinite in MatLab.

Si producano i risultati usando almeno due metodi di interpolazione diversi per le seguenti matrici

- I matrice di numeri casuali di dimensione 32×32
- I matrice binaria di dimensione 64×64 così definita
$$I(i,j) = 150, \quad 21 < i < 43, 21 < j < 43$$
$$I(i,j) = 50. \quad \text{altrimenti}$$
- I immagine a livelli di grigio (usare la funzione Matlab `imread.m` per leggere la immagine. Per esempio: `I = double(imread('cameraman.tif'));`)

Commentare i risultati ottenuti, scrivendo una breve relazione che includa i risultati più significativi.

10. Scrivere una funzione Matlab che:

- legga in input una funzione f e tre numeri reali a, b, Δ
- separi gli zeri di f contenuti nell'intervallo $[a, b]$ usando il metodo della tabulazione con passo Δ
- visualizzi il grafico della funzione f in ogni intervallo individuato al punto precedente e, per ognuno di essi, richieda di inserire da tastiera il numero 1 se la radice è unica, 0 altrimenti. Si etichettino opportunamente i grafici.
- restituisca in output gli intervalli di separazione trovati. La variabile di output I è una matrice con un numero di righe pari al numero di intervalli in cui esiste almeno uno zero di f e un numero di colonne pari a 3. Le prime due colonne di I devono contenere rispettivamente estremo sinistro e destro di ogni intervallo; la terza colonna deve contenere 1 se la radice è unica 0 altrimenti.

Usando la funzione precedente, separare gli zeri delle seguenti funzioni

- $f(x) = \cos(x) - \cos(3.1 x)$ nell'intervallo $[-1, 8]$
- $f(x) = x^3 - 3x^2 + 1$ nell'intervallo $[-5, 5]$
- $f(x) = \sin(\pi x)/x$ nell'intervallo $[-3, 3]$

usando i passi 1, $\frac{1}{2}$ e $\frac{1}{4}$. Si discutano i risultati ottenuti.

11. Scrivere una funzione matlab che effettui la convoluzione tra due segnali. La funzione deve prendere in **input** i due segnali x e y e restituire in **output** il segnale z risultante dall'operazione di convoluzione.

In particolare la funzione deve

- 1) calcolare il numero dei campioni dei due segnali e stamparli a video
- 2) restituire un messaggio di errore se uno dei due parametri di input non è stato assegnato
- 3) calcolare il segnale z dato dal prodotto di convoluzione tra i due segnali in entrata x e y
- 4) stampare a video la lunghezza del segnale z
- 5) disegnare in una stessa finestra grafica i tre segnali x , y e z mediante l'uso del comando subplot
- 6) salvare il grafico ottenuto al punto 5)

Scrivere una breve relazione in cui si spieghi l'algoritmo implementato e in cui si riportino i grafici dei segnali. Includere una breve descrizione sull'operazione di convoluzione e sulla sua importanza in teoria dei segnali. Supponendo che i segnali x e y siano campionati a 1000 Hz, si applichi la funzione di convoluzione ai seguenti segnali

- i) due rettangoli di durata 300 ms
- ii) $x(n) = u(n) - u(n-10)$ e $y(n) = 0.9^n u(n)$ con $u(n) = n$ se $n \geq 0$; $u(n) = 0$ se $n < 0$ (rappresentare i segnali con la funzione stem). I segnali x e y hanno rispettivamente 20 e 40 campioni
- iii) un rettangolo di durata 0.3 s e un segnale esponenziale $y(t) = \exp(-t/T)$ con $T=0.5$ s; considerare l'intervallo di tempo $t = [0 \ 400]$ ms

12. Scrivere una funzione matlab che prenda in **input** una stringa *filename*, un valore di soglia *threshold* e restituisca in **output** l'immagine originale *I*, l'immagine contenente i contorni estratti *Ic* e l'immagine filtrata *If*. In particolare la funzione deve

- leggere l'immagine contenuta in *filename* e memorizzarla in una variabile *I*
- visualizzare l'immagine in scala di grigio
- estrarre i valori dell'immagine al di sopra della soglia *threshold* e creare una nuova immagine *Ic* contenente solo i valori estratti
- visualizzare in un unico grafico *I* e *Ic*
- aggiungere all'immagine *I* del rumore sale e pepe
- implementare un filtro che tolga il rumore aggiunto all'immagine *In* così creata
- visualizzare in uno stesso grafico l'immagine originale *I*, l'immagine rumorosa *In* e l'immagine filtrata *If*

Scrivere una breve relazione sul lavoro svolto. Usare come immagine campione *mri.tif*