- Non si possono consultare libri, note, ed ogni altro materiale o persone durante l'esame ad eccezione delle funzioni Matlab fornite.
- Risolvere i seguenti esercizi con l'ausilio di Matlab.
- La durata del compito è di 90 minuti.
- Questo esame ha 3 domande, per un totale di 30/30 punti.
- Svolgere gli esercizi su fogli protocollo, indicando: nome, cognome, codice persona e data
- Per ciascun esercizio consegnare su webeep un file nominato, ad esempio, "esercizio1.m" con il codice Matlab sviluppato.
- Per utilizzare le funzioni Matlab sviluppate durante il corso e fornite per l'esame, è necessario aggiungere la cartella con il comando `addpath functions2023`.

Esercizio 1 (punti 10)

Si consideri il seguente problema di Cauchy

$$\begin{aligned} & mx''(t) + sx'(t) + kx(t) = f(t) \qquad 0 < t < 10 \\ & x(0) = 0 \\ & x'(0) = 1 \end{aligned} \tag{1}$$

dove $m = 1$, $s = 2$, $k = 10$ e $f(t) = \sin(t)$.

(a) (2 punti) [T] Si scriva il problema di Cauchy (1) come un sistema del prim'ordine.

Soluzione. Il problema (1) consiste in una equazione del second'ordine, che può essere trasformata in *due* equazioni del primo ordine definendo:

$$\begin{cases} \boldsymbol{y}'(t) & = \boldsymbol{F}(t, \boldsymbol{y}) \\ \boldsymbol{y}(0) & = \boldsymbol{y_0} \, . \end{cases}$$

il vettore $\boldsymbol{y}$ è un vettore colonna di due componenti $\boldsymbol{y}(t) = (y_1(t),\, y_2(t))^T$ tale per cui $y_1 = x$ $y_2 = x'$. La funzione $\boldsymbol{F}$ è definita nel seguente modo

$$\boldsymbol{F}(t, \boldsymbol{y}) = \begin{bmatrix} y_2 \\ -\frac{k}{m} y_1 - \frac{s}{m} y_2 + \frac{f(t)}{m} \, , \end{bmatrix}$$

ed il vettore dei valori iniziali è composto da $\boldsymbol{y}_0 = (x(0), x'(0))^T$. Notiamo inoltre che sfruttando la linearità in $\boldsymbol{y}$ il vettore forzante $\boldsymbol{F}$ si può esprimere come:

$$\boldsymbol{F}(t, \boldsymbol{y}) = M \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ f(t) \end{bmatrix}$$

dove

$$M = \begin{bmatrix} 0 & 1 \\ -km^{-1} & -sm^{-1} \end{bmatrix}$$

(b) (3 punti) [T] Introdurre il θ-metodo per la discretizzazione per un sistema di equazioni differenziali ordinarie, compresa la formulazione matriciale associata. Quali metodi otteniamo variando θ? Che proprietà hanno questi metodi?

Soluzione. Il θ-metodo, per un sistema di ODE, ha la seguente espressione

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{h} = \theta \boldsymbol{F}(t^{n+1}, \boldsymbol{u}^{n+1}) + (1-\theta)\boldsymbol{F}(t^n, \boldsymbol{u}^n).$$

dove h è l'ampiezza del passo di discretizzazione temporale e $\theta \in [0, 1]$. Nel caso specifico, può essere scritto come

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{h} = \theta M \boldsymbol{u}^{n+1} + (1-\theta) M \boldsymbol{u}^n + \theta \begin{bmatrix} 0 \\ f(t^{n+1}) \end{bmatrix} + (1-\theta) \begin{bmatrix} 0 \\ f(t^n) \end{bmatrix}.$$

Per $\theta = 0$ otteniamo il metodo di Eulero Esplicito, convergente con ordine 1 e condizionatamente assolutamente stabile; per $\theta = 0.5$ otteniamo il metodo di Crank-Nicolson, convergente con ordine 2, e per $\theta = 1$ otteniamo il metodo di Eulero Implicito, di ordine 1. Gli ultimi due metodi, impliciti, sono incondizionatamente assolutamente stabili.

(c) (3 punti) [M] Risolvere il sistema precedente mediante il metodo di Eulero Esplicito implementato nella funzione `eulero_avanti_sis` con i seguenti passi temporali $\Delta t = [0.1, 0.01, 0.001]$. Rappresentare graficamente le soluzioni (in particolare $y_1 = x$) e quindi commentare quanto ottenuto sulla base della teoria.

Suggerimento: per definire la forzante per `eulero_avanti_sis` usare una sintassi del tipo

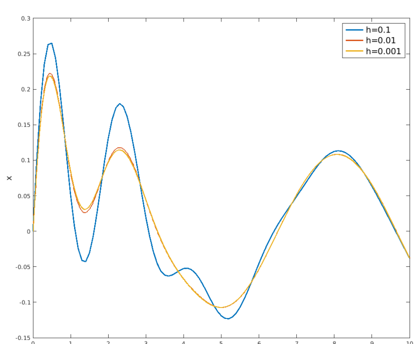
```
yp= @(t, y) [... ; ...]
```

Soluzione. Definiamo i dati utili per la soluzione del problema:

```
addpath functions2023
close all
clear all

m=1;
s=2;
k=10;

M=[0 1;-k/m -s/m];
f=@(t) sin(t);

yp=@(t,y) M*[y(1);y(2)] + [0; m^-1*f(t)];
```

Definito un vettore di passi temporali ripetiamo il calcolo con il metodo di Eulero Esplicito al variare di h e rappresentiamo le soluzioni, sovrapposte in un grafico:

```
for h=[0.1 0.01 0.001 ]
   [t_h,u_h]=eulero_avanti_sis(yp,0,10,[0;1],h);
   plot(t_h, u_h(1,:),'linewidth',2)
   hold on
end
legend('h=0.1','h=0.01','h=0.001','FontSize',14)
xlabel('t','FontSize',14)
ylabel('x','FontSize',14)
```



Possiamo notare che per il passo più grande la soluzione è molto inaccurata, mentre è sostanzialmente uguale per $h = 0.01$ e $h = 0.001$. Inoltre si nota che, dopo un transitorio iniziale, la soluzione si "sincronizza" con la forzante sinusoidale.

(d) (2 punti) [M] Risolvere lo stesso problema utilizzando il metodo `ode45` implementato in Matlab. Rappresentare la soluzione ottenuta evidenziando l'ampiezza dei passi temporali utilizzati, e commentare il risultato.

Soluzione. Forniamo al metodo solo l'intervallo temporale e i dati iniziali: l'ampiezza del passo sarà decisa adattivamente.

```
[t_h,u_h]=ode45(yp,[0,10],[0;1]);
figure
plot(t_h, u_h(:,1),'linewidth',2)
hold on
plot(t_h, 0*u_h(:,1),'b.')
length(u_h)
```

Notiamo nel grafico che la spaziatura dei punti non è uniforme, il passo temporale è più piccolo in corrispondenza dell'inizio della simulazione e nei punti in cui si inverte il segno della velocità. Inoltre, notiamo che con soli 149 passi si ottiene una soluzione comparabile con quella ottenuta con 1000 step di Eulero Esplicito.

Esercizio 2 (punti 10)
Si consideri il seguente integrale

$$I = \int_0^{\frac{3}{2}} f(x)dx \quad \text{per} \quad f(x) = \sec(x)$$

la cui primitiva dell'integranda è data da

$$F(x) = \ln|\tan(x) + \sec(x)|$$

(a) (4 punti) [T] Introdurre il metodo di quadratura del punto medio e di Simpson e la loro versione composita. Qualè l'ordine di accuratezza per tali metodi? E il loro grado di esattezza?

Soluzione. Per la formula del punto medio l'integrale viene approssimato da

$$I \approx I_{pm} = (b-a)f\,(m)$$

dove m denota il punto medio dell'intervallo di integrazione $[a, b]$, dato da $m = 0.5(a + b)$, l'errore per la formula del punto medio è dato da

$$|E_{pm}| \leq \frac{(b-a)^3}{24} \max_{x \in I} |f''(x)|\,.$$

Mentre la formula di Simpson, in cui approssimiamo I utilizzando il valore della funzione in tre punti, è data da

$$I \approx I_s = \frac{b-a}{6}\left[f(a) + 4f\,(m) + f(b)\right],$$

dove $m = 0.5(a + b)$, il cui errore è dato da

$$|E_s| \leq \frac{(b-a)^5}{16 \cdot 180} \max_{x \in [a,b]} \left|f^{(iv)}(x)\right|.$$

Nell'integrazione composita suddividiamo l'intervallo $[a, b]$ in n sotto-intervalli, dove, per $i = 1, \ldots, n$, ogni intervallo è dato da $I_i = [x_{i-1}, x_i]$ con $x_i = a + iH$ e dove $H = (b - a)/n$. Otteniamo così la formula del punto medio composito in cui il valore dell'integrale è calcolato come

$$I^c_{pm} = H \sum_{i=1}^{n} f\left(\frac{x_{i-1} + x_i}{2}\right).$$

e la formula di Simpson composita risulta data da

$$I^c_s = \frac{H}{6} \sum_{i=1}^{n} \left[f(x_{i-1}) + 4f\left(\frac{x_{i-1} + x_i}{2}\right) + f(x_i)\right].$$

L'errore del punto medio composito è dato da $\left|E^c_{pm}\right| \leq cH^2$ e per Simpson abbiamo $|E^c_s| \leq cH^4$. L'ordine di accuratezza per il punto medio è pari a 2 mentre per Simpson pari a 4. Mentre il grado di esattezza è pari a 1 e 3 rispettivamente.

(b) (4 punti) [M] Usando le funzioni `pmedcomp` e `simpcomp` calcolare un'approssimazione di I per le seguenti suddivisioni dell'intervallo di integrazione $N = [10, 20, 40, 80]$. Calcolare l'errore ottenuto e rappresentarlo in scala logaritmica. Commentare i risultati ottenuti alla luce della teoria.

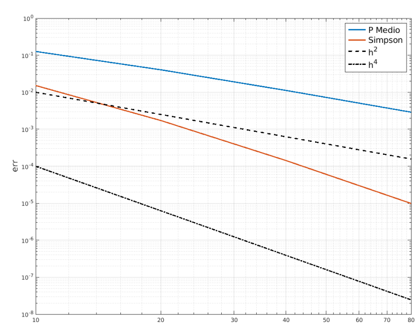
Soluzione. Definiamo la funzione, l'intervallo e l'integranda per poter calcolare il valore esatto dell'integrale.

```
close all
f=@(x) sec(x);
F=@(x) log(tan(x)+sec(x));
a=0;
b=3/2;
I_ex=F(b)-F(a);
```

Quindi calcoliamo il valore dell'integrale per i diversi valori N_i grazie a un ciclo:

```
N=[10 20 40 80];
err_pm=[];
err_s=[];

for n=N
    Ipm = pmedcomp( a, b, n, f );
    err_pm=[err_pm abs(Ipm-I_ex)];
    Is = simpcomp( a, b, n, f );
    err_s=[err_s abs(Is-I_ex)];
end
```

e otteniamo il seguente grafico



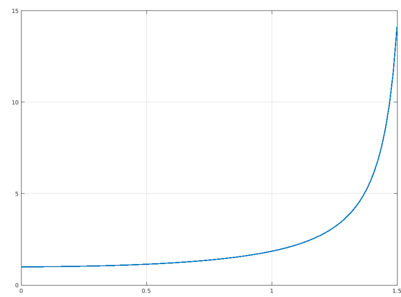
dove possiamo osservare che, come previsto dalla teoria, l'errore decresce quadraticamente con il metodo del punto medio, e con ordine 4 nel caso del metodo di Simpson.

(c) (2 punti) [T] Rappresentare graficamente l'integranda $f(x)$, proporre una strategia per di scomposizione dell'intervallo di integrazione in modo da aumentare l'accuratezza mantenendo lo stesso numero di divisioni.

Soluzione. Con i seguenti comandi rappresentiamo l'andamento della funzione:

```
figure
plot(linspace(a,b,100),f(linspace(a,b,100)),'linewidth
    ',2 )
```

ottenendo:



Notiamo che la funzione ha inizialmente una derivata molto bassa, mentre in prossimità dell'estremo $b = \frac{3}{2}$ l'andamento ha una derivata molto maggiore. Una possibile strategia potrebbe essere quella di applicare l'integrazione composita con una risoluzione diversa nelle due parti, oppure addirittura di adattare l'ampiezza degli intervalli di integrazione in modo continuo per riflettere l'andamento della soluzione. Implementiamo la prima, più semplice, soluzione. Consideriamo il caso di $N = 80$ intervalli, in cui il metodo del punto medio fornisce un errore pari a 0.0029. Calcoliamo $I = I_1 + I_2$ dove I_1 è l'integrale approssimato nell'intervallo $(0, 1.25)$ e I_2 nell'intervallo $(1.25, 1.5)$. Usiamo $N_1 = 40$ intervalli per il calcolo di I_1, e $N_2 = 40$ per I_2:

```
I_1= pmedcomp( a, 1.25, 40, f );
I_2= pmedcomp( 1.25, b, 40, f );
abs(I_1+I_2-I_ex)
```

Otteniamo un errore pari a $6.9620e-04$ con lo stesso numero di intervalli.

Esercizio 3 (punti 10)

Assumendo $c > 0$, si consideri la seguente equazione di conservazione

$$\begin{cases} \partial_t c + \partial_x(0.5c^2) = 0 & x \in (0,5),\, t \in (0,T] \\ c(0,x) = e^{-(x-2)^2} & x \in (0,5) \\ c(t,0) = 0 & t \in (0,T] \end{cases}$$

dove il tempo finale è pari a $T = 1$.

(a) (3 punti) [T] Verificare se il flusso numerico Upwind è applicabile per la discretizzazione dell'equazione proposta con il metodo dei volumi finiti. Scegliere il valore $h = 0.125$ costante per l'ampiezza delle celle. Calcolare quindi il massimo Δt ammesso e chiamare Δt_{max} tale valore.

> **Soluzione.** Nel caso in esame abbiamo $f(c) = \frac{1}{2}c^2$ e $f'(c) = c$. In base ai dati iniziali forniti il valore minimo e massimo di c sono rispettivamente $c_m = c(5) = 1.2341e - 04$, $c_M = 1$ e, per tali valori, la derivata prima della funzione flusso è sempre non negativa qiundi il flusso upwind è applicabile. Il massimo della derivata $f'(c)$ si ottiene in corrispondenza di $c = 1$ e vale $\max_{c\in(c_m,c_M)} |f'(c)| = 1$. Per soddisfare la condizione CFL dobbiamo quindi garantire che $\Delta t \max_{c\in(c_m,c_M)} |f'(c)| < h$. Con i valori scelti si ottiene $\Delta t_{max} = 0.125$.

(b) (3 punti) [M] Sia $N = T/\Delta t_{max}$, risolvere il problema utilizzando la function `fvsolve` utilizzando il metodo di Upwind per N, $N-2$ e $2N$ passi temporali (attenzione: calcolare i Δt corrispondenti!) e rappresentare le soluzioni ottenute usando la function `xtplot`. Commentare cosa si osserva.

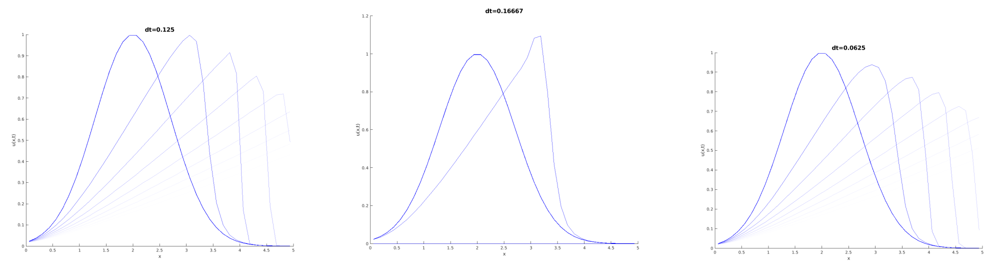
> **Soluzione.** Il numero di intervalli N si ottiene come $N = 1/0.125 = 8$. Definiamo i dati necessari per la soluzione numerica,

```
dtlim=h/df(1)
N=T/dtlim;

for n=[N, N-2, N*2]
 [xc, t, u] = fvsolve(u0, f, df, L, 10, h, T/n, '
    UPWIND');
 figure
 xtplot(xc,t,u,'fade')
 title(strcat('dt=',num2str(T/n)),'FontSize',14)
end
```

> Impostiamo quindi un ciclo per calcolare la soluzione con il numero di intervalli richiesto:

```
for n=[N, N-2, N*2]
 [xc, t, u] = fvsolve(u0, f, df, L, 10, h, T/n, '
    UPWIND');
 figure
 xtplot(xc,t,u,'fade')
 title(strcat('dt=',num2str(T/n)),'FontSize',14)
end
```

ottenendo i tre grafici riportati:



Osserviamo che per il passo temporale "limite" la soluzione è corretta e si osserva la formazione di uno shock; per un passo temporale leggermente superiore al massimo consentito dalla CFL si ha un andamento non fisico che porta a un arresto del solver; per passi temporali minori si ha una soluzione corretta, con una maggiore diffusine numerica.

(c) (2 punti) [T] Si descriva il flusso numerico di Godunov e se ne discutano le proprietà.

Soluzione. L'idea per introdurre il flusso di Godunov è quella di partire da una soluzione costante a tratti al tempo t_n, e di risolvere problemi di Riemann ad ogni interfaccia fra due celle vicine. Per calcolare il flusso di Godunov fra le celle i e $i+1$ consideriamo che, a seconda dei valori di c_i^n e c_{i+1}^n e della "forma" del flusso, possiamo avere uno shock, una rarefazione o una combinazione dei due; questo determina il valore della soluzione c^* nel nodo $i+\frac{1}{2}$ nell'intervallo di tempo $[t_n, t_{n+1}]$. Utilizzando c^* possiamo valutare il flusso numerico all'interfaccia.

Il flusso di Godunov risultante si può esprimere in modo sintetico con la seguente definizione:

$$F^G_{i+\frac{1}{2}}(c_i, c_{i+1}) = \begin{cases} \min f(\xi),\ \xi \in [c_i, c_{i+1}] & \text{se } c_i \leq c_{i+1} \\ \max f(\xi),\ \xi \in [c_{i+1}, c_i] & \text{se } c_i \geq c_{i+1} \end{cases}$$

Osserviamo che se f è monotono questa definizione coincide con il flusso upwind.

(d) (2 punti) [M] Si calcoli la soluzione del problema proposto utilizzando ora il flusso numerico di Godunov e un opportuno passo temporale.

Soluzione. Scegliamo ad esempio un passo $\Delta t = 0.9\Delta t_{max}$:

```
[xc, t, u] = fvsolve(u0, f, df, L, 10, h, 0.9*dtlim,
   'GODUNOV');
figure
xtplot(xc,t,u,'fade')
```

La soluzione ottenuta è simile a quella ottenuta con il flusso upwind.