- 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 **??** domande, per un totale di **??**/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, è necessario aggiungere la cartella con il comando `addpath functions2022`.

Esercizio 1 (punti **??**)

Si consideri il seguente problema differenziale di tipo diffusione-reazione

$$\begin{cases} -u''(x) + 3u(x) = \cos(\pi x), & 0 < x < 2, \\ u(0) = 0, \qquad u(2) = 0. \end{cases}$$

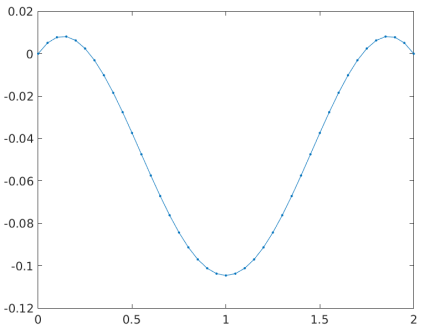
(a) (3 punti) Si calcoli la soluzione numerica del problema con elementi finiti lineari su una griglia uniforme con diversi passi $h = 0.2, 0.1, 0.05$. Per ciascun passo, si riporti il valore della soluzione valutata nel centro del dominio $x = 1$. [M]

Soluzione. Ecco un possibile script

```
% a
clear
close all
clc

addpath functions2021
format long

f = @(x) cos(pi*x);

for h = [0.2, 0.1, 0.05]
    [X, U] = direlliptic(1, 0, 3, f, 2, h);
    Uinterp = @(x) interp1(X, U, x);
```

```
    Uinterp(1)
end

plot(X, U, '.-')
saveas(gcf, "es1_sol_a.png")
```

La soluzione in $x = 1$ vale -0.108453283303437 ed è riportata nella seguente figura



(b) (3 punti) Si discuta sotto quali condizioni sulla matrice, un sistema lineare può essere risolto utilizzando il metodo del gradiente coniugato e si verifichi numericamente che il sistema risolto nel punto precedente soddisfi tali condizioni. [T+M]

Soluzione. Sia $A \in \mathbb{R}^{n\times n}$ una matrice simmetrica e definita positiva, allora il metodo del gradiente coniugato converge alla soluzione del sistema $A\boldsymbol{x} = \boldsymbol{b}$, per ogni $\boldsymbol{x}^0 \in \mathbb{R}^n$ in al più n iterazioni. Inoltre, vale la stima

$$\left\|\boldsymbol{e}^k\right\|_A \leq \frac{2c^k}{1+c^{2k}} \left\|\boldsymbol{e}^0\right\|_A \quad \text{dove} \quad c = \frac{\sqrt{\text{cond}(A)}-1}{\sqrt{\text{cond}(A)}+1}.$$

La forma bilineare associata al problema risulta coerciva e quindi la matrice è sicuramente definita positiva, inoltre risulta anche simmetrica che implica quindi la simmetria della matrice.

Per il problema in questione tale condizione è verificata come riportata dallo script `direlliptic`.

(c) (4 punti) Si ripeta il punto a. dopo avere modificato lo script `direlliptic.m` in modo che il sistema lineare sia risolto con il metodo del gradiente coniugato e si riporti il numero di iterazioni necessarie per raggiungere la convergenza, a meno di una tolleranza pari a 10^{-6}. Si commenti il risultato ottenuto rispetto alle proprietà di convergenza del metodo del gradiente coniugato. [M+T]

Soluzione. Lo script modificato è riportato di seguito

```
function [X,U] = direllipticiter(a,b,c,fun,L,h)

% [X,U] = direlliptic(a,b,c,fun,L,h)
%
% Risolve il problema ai limiti (problema di Dirichlet
    omogeneo)
% con coefficienti costanti a, b, c
%
% -au'' + bu' + cu = f(x), in [0,L],
% u(0) = u(L) = 0,
%
% con il metodo degli elementi finiti (lineari).
%
% INPUT:
% a = coeff. di diffusione;
% b = coeff. di trasporto;
% c = coeff. di reazione;
% fun = function handle per il termine noto;
% L = lunghezza dominio 1D;
% h = passo di discretizzazione spaziale.
%
% OUTPUT:
% X = vettore che contiene i nodi della griglia;
% U = vettore che contiene la soluzione approssimata
   sui nodi della
% griglia.

% nodi di bordo
x0 = 0;
xL = L;

% nodi interni
xin = [x0+h:h:xL-h]';
N = length(xin);

% matrice di rigidezza A
d0  = ones(N,1);
d1  = ones(N-1,1);

A = a/h  * (2*diag(d0) - diag(d1,1) - diag(d1,-1))+ ...
    b    * (1/2*diag(d1,1) - 1/2*diag(d1,-1))+ ...
```

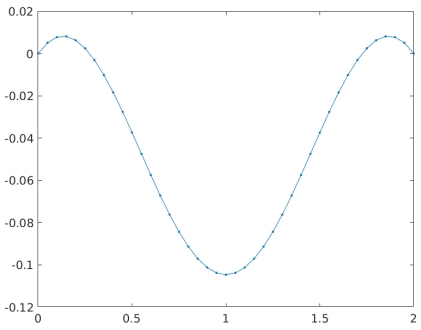
```
    c*h   * (2/3*diag(d0) + 1/6*diag(d1,1) + 1/6*diag(
       d1,-1));

% termine noto f
f = h*fun(xin);
f(end) = f(end);


% Check sulla simmetria di A (garantita, se b = 0 -->
   NO TRASPORTO)
if A == A'
   fprintf('A risulta simmetrica \n')
else
   fprintf('A non risulta simmetrica \n')
end

% Check sulla positivita di A (garantita, se a > 0, c
   >= 0 (condizioni per soddisfare coercivita di a(u,v
   )))
if min(eig(A))>0
    fprintf('A risulta definita positiva \n')
else
    fprintf('A non risulta definita positiva \n')
end

% risoluzione sistema lineare (con backslash di MATLAB
   )
% u = A\f;
[u,iter,xdif,err]=gradconj(A,f,f.*0,10000,1e-6);
disp('iterazioni')
iter

% aggiunta dei bordi per i nodi di griglia e per la
   soluzione
X = [x0;xin;xL];
U = [0;u;0];

end
```

Riprendiamo ora il punto a

```
%% c
```

```
for h = [0.2, 0.1, 0.05]
    [X, U] = direllipticiter(1, 0, 3, f, 2, h);
    Uinterp = @(x) interp1(X, U, x);
    Uinterp(1)
end

plot(X, U, '.-')
saveas(gcf, "es1_sol_c.png")
```

La soluzione in $x = 1$ vale -0.105371225860506 ed è riportata nella seguente figura



Si richiedono 5, 10 e 20 iterazioni rispettivamente per la risoluzione del sistema lineare per i differenti passi di discretizzazione.

Tale numero di iterazioni è coerente con la teoria infatti la dimensione della matrice è pari a 5, 10 e 20 rispettivamente per ogni passo di discretizzazione.

Esercizio 2 (punti **??**)

Si vuole integrare la funzione $f(x) = x^6 - 3x + 2$ sull'intervallo [0,2].

(a) (3 punti) Si approssimi l'integrale $I = \int_0^2 f(x)\mathrm{d}x$ utilizzando la formula di quadratura dei trapezi composita con un numero di sottointervalli variabile tra 1 e 10, si valuti analiticamente l'integrale esatto e si riporti il valore dell'errore al variare del numero di sottointervalli. [M]

Soluzione. Una primitiva della funzione f è data da

$$F(x) = \frac{1}{7}x^7 - \frac{3}{2}x^2 + 2x + c$$

Ecco un possibile script

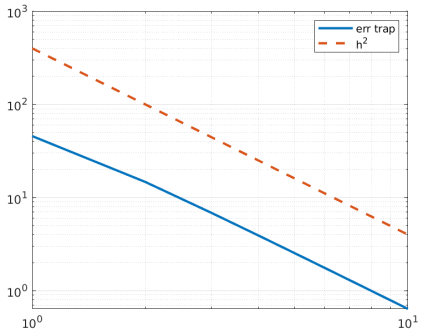
```
% a
clear
close all
clc

addpath functions2021

f = @(x) x.^6 - 3*x + 2;

Fex = @(x) x.^7/7 - 3*x.^2/2 + 2*x;
a = 0;
b = 2;
Iex = Fex(b) - Fex(a);
err_trap = [];
NN = 1:10;
for i = NN
    I = trapcomp(a, b, i, f);
    err_trap = [err_trap, abs(Iex - I)];
end

figure
loglog(NN, err_trap, 'linewidth', 2)
hold on
loglog(NN, (20./NN).^2, '--', 'linewidth', 2)
legend('err trap', 'h^2')
grid on
saveas(gcf, "es2_sol_a_err.png")
```

L'errore calcolato risulta

```
45.714285714285715
```

```
14.714285714285715
6.851917172904830
3.917410714285715
2.526061714285721
1.761382193480962
1.297263166586326
0.994803292410715
0.786877654563813
0.637869714285717
```

Graficamente risulta



(b) (3 punti) Si approssimi l'integrale $I = \int_0^2 f(x)\mathrm{d}x$ utilizzando l'integrazione gaussiana con un numero di nodi di Gauss variabile tra 2 e 5 e si riporti il valore dell'errore al variare del numero di nodi. [M]

Soluzione. Ecco un possibile script

```
%% b
err_gauss = []
for i = 2:5
    I = quadgauss(a, b, f, i);
    err_gauss = [err_gauss, abs(Iex - I)];
end
```

L'errore calcolato risulta

```
2.878306878306878
0.045714285714286
3.552713678800501e-15
7.105427357601002e-15
```

(c) (4 punti) Si commentino i risultati ottenuti nei punti precedenti alla luce delle proprietà di accuratezza ed esattezza delle formule di quadratura considerate. [T]

Soluzione. La formula del trapezio composita ha grado di esattezza pari a 2, non integra quindi esattamente la funzione f assegnata. Inoltre l'errore decade come H^2, confermato dal grafico riportato nel punto (a).

Per la formula di quadratura di Gauss, il grado di esattezza è pari a $m = 2n + 1$, avendo $n + 1$ nodi, ovvero per numero di nodi che variano da 2 a 5 risulta rispettivamente $[3, 5, 7, 9]$. Come riportato dal calcolo dell'errore nel punto precedente si nota che per 4 e 5 nodi l'errore commesso è zero, mentre per i primi due casi l'errore è non nullo.

Esercizio 3 (punti **??**)

Si consideri la funzione $f(x) = x^3(10-x)^4 \sin(\pi x)$ nell'intervallo [0,10].

(a) (4 punti) Si calcoli l'interpolante spline cubica naturale della funzione $f(x)$ su un numero di sotto-intervalli variabile con $n = 2^i$ con $i = 1, \ldots, 6$. Calcolare l'errore in norma infinito al variare del numero di sotto-intervalli valutando la spline in 1000 punti nell'intervallo considerato. [M]

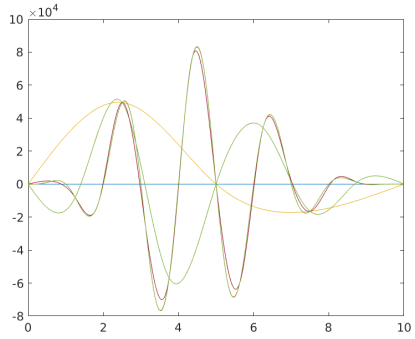
Soluzione. Ecco un possibile script

```
% a
clear
close all
clc

addpath functions2021

f = @(x) x.^3.*((10 - x).^4).*sin(pi*x);

a = 0;
b = 10;
xx = linspace(a, b, 1000);

err = [];
NN = 2.^(1:6);
for i = NN
    x = linspace(a, b, i+1);
    yy = spline_nat(x, f(x), xx);
    plot(xx, yy)
    hold on
    plot(xx, f(xx))
    err = [err, norm(f(xx) - yy, Inf)];
end
saveas(gcf, "es3_sol_a.png")

figure
loglog(b./NN, err, 'linewidth', 2)
hold on
loglog(b./NN, (20*b./NN).^2, '--', 'linewidth', 2)
loglog(b./NN, (20*b./NN).^3, '--', 'linewidth', 2)
loglog(b./NN, (20*b./NN).^4, '--', 'linewidth', 2)
legend('1', 'h^2', 'h^3', 'h^4')
grid on
saveas(gcf, "es3_sol_a_err.png")
```

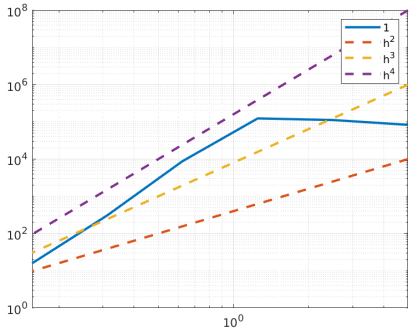
La norma infinito risulta quindi data da

```
      1.0e+05 *
0.8340    1.1230    1.2387    0.0879    0.0031
   0.0002
```

La rappresentazione grafica dell'interpolante è data nella seguente figura



Mentre l'errore calcolato è rappresentabile graficamente come



(b) (2 punti) Si riporti la stima teorica dell'ordine di convergenza per le spline cubiche e si discutano i risultati ottenuti nel punto precedente. [T]

Soluzione. L'ordine di convergenza per le spline cubiche è il seguente

$$\max_{x\in I} \left| f^{(r)}(x) - s_3^{(r)}(x) \right| \leq CH^{4-r}$$

dove r è il grado di derivazione. Nel nostro caso stiamo considerando $r = 0$ e quindi ci aspettiamo un andamento quadratico dell'errore. Ciò è confermato dal grafico del punto (a) in cui l'errore calcolato tende ad essere parallelo alla curva H^4.

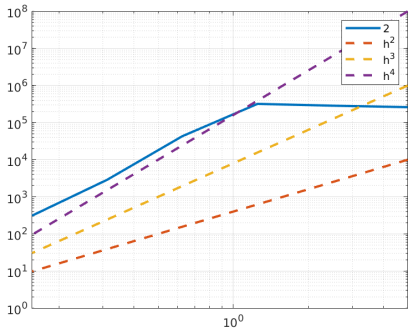
(c) (4 punti) Si utilizzino le differenze finite centrate per approssimare la derivata prima delle spline sugli stessi 1000 punti (limitatamente ai punti interni). Si valuti l'errore in norma infinito rispetto alla derivata prima della funzione f e si discuta

il risultato. [M+T]

Soluzione. Ecco un possibile script

```
%% c
df = @(x) 3*x.^2.*sin(pi*x).*(x - 10).^4 + ...
          4*x.^3.*sin(pi*x).*(x - 10).^3 + ...
          x.^3*pi.*cos(pi*x).*(x - 10).^4;
derr = [];
NN = 2.^(1:6);
for i = NN
    x = linspace(a, b, i+1);
    yy = spline_nat(x, f(x), xx);
    dfh = (yy(3:end) - yy(1:end-2))./(2*b/999);
    derr = [derr, norm(df(xx(2:end-1)) - dfh, Inf)];
end

figure
loglog(b./NN, derr, 'linewidth', 2)
hold on
loglog(b./NN, (20*b./NN).^2, '--', 'linewidth', 2)
loglog(b./NN, (20*b./NN).^3, '--', 'linewidth', 2)
loglog(b./NN, (20*b./NN).^4, '--', 'linewidth', 2)
legend('2', 'h^2', 'h^3', 'h^4')
grid on
saveas(gcf, "es3_sol_c_err.png")
```

L'errore calcolato è rappresentabile graficamente come



Utilizzando il risultato di convergenza presentato al punto precedente, ci aspettiamo una convergenza dell'ordine di H^3 avendo $r = 1$. Il grafico mostrato conferma tale risultato.