

- 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 (caricare le function fornite per l'esame solo se modificate).
- 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 la seguente matrice

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{bmatrix}$$

che si può costruire con il comando Matlab `pascal(n)` dove n è la dimensione della matrice.

- (a) (3 punti) [M+T] Enunciare la condizione necessaria e sufficiente per l'esistenza e unicità della fattorizzazione LU e verificare (con opportuni comandi) che è soddisfatta per la matrice A di dimensione $n = 5$.

Soluzione. Data una matrice $A \in \mathbb{R}^{n \times n}$ non singolare la fattorizzazione LU esiste ed è unica se e solo se tutte le sotto-matrici principali A_i di ordine $i = 1 \dots n - 1$ sono non singolari. Possiamo calcolare il determinante delle sottomatrici con il seguente ciclo:

```
close all
clear all
clc
```

```
addpath functions2023
```

```
%% a)
```

```
n=5;
```

```
A=pascal(n);
```

```
for i=1:n
```

```
    D(i)=det(A(1:i,1:i));
```

```
end
```

```
if (all(D~=0))
```

```
    disp('la fattorizzazione LU (senza pivoting) esiste  
        ')
```

```
end
```

Chiedendo che tutti i valori salvati nel vettore D siano diversi da zero verifichiamo che tutte le sotto-matrici principali e la matrice A stessa siano non-singolari (condizione verificata).

- (b) (3 punti) [M] Data la soluzione esatta $\mathbf{x} = [1, 1 \dots, 1]^T$ costruire il termine noto \mathbf{b} e risolvere il sistema lineare $A\mathbf{x} = \mathbf{b}$ utilizzando i) la fattorizzazione LU ii) i metodi di sostituzione in avanti e all'indietro implementati nelle funzioni fornite. Verificare se è stato effettuato il pivoting.

Soluzione. Definita la soluzione esatta il termine noto si calcola come $\mathbf{b} = A\mathbf{x}$:

```
%% b)
```

```
x=ones(n,1);
```

```
b=A*x;
```

```
[L,U,P]=lu(A);
```

```
if any(any(P-eye(n)))
```

```
    disp('il pivoting e stato effettuato')
```

```
end
```

```
y=fwsb(L,P*b);
```

```
xn=bksb(U,y)
```

Ottenendo i seguenti risultati

```
la fattorizzazione LU (senza pivoting) esiste
```

```
il pivoting e stato effettuato
```

```
xn =
```

```
    1
```

```
    1
```

1
1
1

Abbiamo effettuato la fattorizzazione chiedendo in output la matrice di permutazione; confrontandola con la matrice identità si può dedurre che il pivoting, anche se non necessario, è stato effettuato. Infine, abbiamo risolto i problemi triangolari, ottenendo la soluzione numerica \mathbf{x}_n . In questo caso la soluzione numerica è identica alla soluzione esatta.

- (c) (4 punti) [M+T] Ripetere i passaggi al punto precedente per matrici di dimensione $n = 10, 15$ e 20 . Per ognuno dei casi calcolare la norma dell'errore *relativo* e il condizionamento della matrice e rappresentarli su due grafici in scala logaritmica. Commentare i risultati alla luce della teoria.

Soluzione. Ripetiamo i passi precedenti per diversi valori di n :

```
%% c)

N=[10 15 20];
err=[];
K=[];

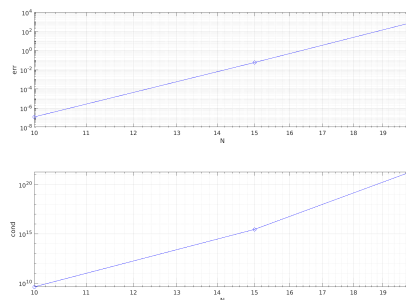
for n=N
    A=pascal(n);
    x=ones(n,1);
    b=A*x;
    [L,U,P]=lu(A);
    y=fwsub(L,P*b);
    xn=bksub(U,y);
    err=[err norm(x-xn)/norm(x)];
    K=[K cond(A)];
end
err
K
subplot(2,1,1)
loglog(N,err,'bo-')
grid on
xlabel('N')
ylabel('err')
subplot(2,1,2)
loglog(N,K,'bo-')
grid on
```

```
xlabel('N')
ylabel('cond')
saveas(gcf, "es1_sol_c.png")
```

Ottenendo i seguenti risultati

```
err =
    1.0e+03 *
    0.0000    0.0001    1.2977
K =
    1.0e+21 *
    0.0000    0.0000    2.2465
```

Ed il seguente grafico



Dal grafico ottenuto osserviamo che per matrici di dimensioni maggiori l'errore cresce fino a superare il 100% e questo si spiega con l'aumento del numero di condizionamento della matrice $K(A) > 10^{20}$. Non è quindi garantito che gli errori di arrotondamento commessi durante la fattorizzazione e il calcolo della soluzione si mantengano bassi, infatti in questo caso sono amplificati inquinando la soluzione numerica.

Esercizio 2 (punti 10)

Si consideri il seguente integrale

$$I = \int_a^b f(x) dx \quad \text{per} \quad f(x) = \frac{1}{\sqrt{x^2 - 1}}$$

la cui primitiva dell'integranda è data da

$$F(x) = \log \left| x + \sqrt{x^2 - 1} \right| + c$$

- (a) (4 punti) [T] Introdurre il metodo di quadratura del punto medio e di Simpson e la loro versione composita. Qual è 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

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

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

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

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

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

Nell'integrazione composita suddividiamo l'intervallo $[a, b]$ in n sotto-intervalli, dove, per $i = 1, \dots, n$, 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

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

e la formula di Simpson composita risulta data da

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

L'errore del punto medio composito è dato da $|E_{pm}^c| \leq cH^2$ e per Simpson abbiamo $|E_s^c| \leq cH^4$. 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 considerando l'intervallo di integrazione compreso fra $a = 1.02$ e $b = 2$, e suddividendolo in $N = [10, 20, 40, 80]$ sotto-intervalli. 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.

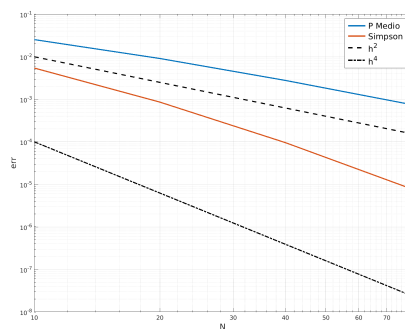
```
close all
f=@(x) (x.^2-1).^(-0.5);
F=@(x) log(abs(x+sqrt(x.^2-1)));
a=1.02;
b=2;
I_ex=F(b)-F(a);
```

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.

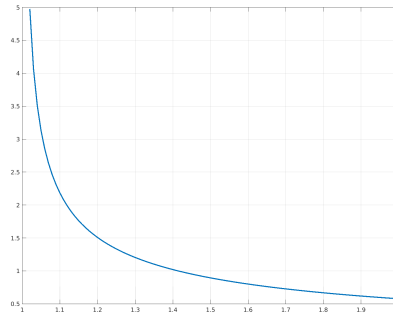
- (c) (2 punti) [T] Rappresentare graficamente l'integranda $f(x)$ nell'intervallo indicato, e proporre una strategia per la scomposizione dell'intervallo di integrazione in modo

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 in prossimità dell'estremo $a = 1.02$ la funzione ha una derivata molto alta (in valore assoluto). Una possibile strategia potrebbe essere quella di applicare l'integrazione composta con una risoluzione diversa nella parte sinistra del dominio, 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 0.0008. Calcoliamo $I = I_1 + I_2$ dove I_1 è l'integrale approssimato nell'intervallo $(1.02, 1.2)$ e I_2 nell'intervallo $(1.2, 2)$. Usiamo $N_1 = 40$ intervalli per il calcolo di I_1 , e $N_2 = 40$ per I_2 :

```
I_1= pmedcomp( a, 1.2, 40, f );  
I_2= pmedcomp( 1.2, b, 40, f );  
abs(I_1+I_2-I_ex)
```

Otteniamo un errore pari a $1.6398e - 04$ con lo stesso numero di intervalli.

Esercizio 3 (punti 10)

Si consideri il seguente problema

$$\begin{cases} -u'' = f(x) & 0 < x < 1, \\ u(0) = 0 \\ u'(1) = 4e^2 \end{cases}$$

- (a) (1 punto) [T] Determinare f tale per cui $u(x) = e^{2x}x^2$ è soluzione esatta del problema e verificare che tale soluzione effettivamente rispetta le condizioni al bordo.

Soluzione. Calcoliamo le derivate della funzione u fornita

$$\begin{aligned} u'(x) &= 2e^{2x}x^2 + 2e^{2x}x \\ u''(x) &= 4e^{2x}x^2 + 4e^{2x}x + 2e^{2x} + 4e^{2x}x = 2e^{2x}(2x^2 + 4x + 1) \end{aligned}$$

verifichiamo inoltre che la soluzione esatta fornita soddisfa le condizioni al bordo

$$u(0) = 0 \quad \text{e} \quad u'(1) = 4e^2$$

quindi effettivamente u è soluzione dell'equazione fornita dove

$$f(x) = -2e^{2x}(2x^2 + 4x + 1).$$

- (b) (4 punti) [T] Introdurre brevemente l'approssimazione del problema con il metodo degli elementi finiti in spazio e derivarne l'espressione matriciale considerando le condizioni al contorno del problema in esame.

Soluzione. Dato che abbiamo una condizione di Dirichlet nel bordo sinistro, in $x = 0$, consideriamo lo spazio funzionale $V = \{v \in H^1(\Omega) : v(0) = 0\}$ e prendiamo delle funzioni test $v \in V$; la forma debole del problema è: trovare $u(x) \in V$ tale che

$$a(u, v) = F(v) \quad \forall v \in V.$$

dove $a(u, v) = \int_0^1 u'v' dx$ e $F(v) = \int_0^1 fv - gv(1)$. Nel nostro caso $f = -2e^{2x}(2x^2 + 4x + 1)$ e $g = 4e^2$. Notiamo che il termine di bordo è nullo sul bordo sinistro mentre la condizione di Neumann sul bordo destro è inclusa nel funzionale.

La formulazione discreta del problema è ottenuta scegliendo un sotto-spazio finito dimensionale $V_h \subset V$ per cui possiamo considerare una base $\{\phi_j\}$ per V_h di funzioni linearmente indipendenti. Nel nostro caso, data una griglia di $N + 2$ nodi in cui $x_0 = 0$, $x_{N+1} = 1$, e spaziatura $h = \frac{b-a}{N+1}$ la base è formata dalle

funzioni “a capanna” ϕ_j con $j = 1, \dots, N + 1$; infatti, considerando lo sviluppo

$$u_h(x) = \sum_{j=1}^{N_h} u_j \phi_j(x)$$

le incognite sono i valori u_j in tutti i nodi eccetto il primo, per un totale di $N + 1$ incognite. Sfruttando la bi-linearità della forma a , il fatto che il dominio non dipende dal tempo e la linearità del funzionale F otteniamo un sistema di $N + 1$ equazioni

$$A\mathbf{u} = \mathbf{f}$$

dove la matrice di rigidezza A

$$A \in \mathbb{R}^{N+1 \times N+1} : \quad a_{ij} = a(\phi_j, \phi_i)$$

In particolare otteniamo la seguente struttura:

$$A = \frac{1}{h} \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & \dots & 0 & 0 & -1 & 2 & -1 \\ 0 & \dots & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

mentre il termine noto \mathbf{f} contiene, nell'ultima riga, il contributo della condizione di Neumann.

- (c) (2 punti) [M] Si consideri una griglia di ampiezza uniforme $h = \frac{1}{10}$. Si risolva il problema con il metodo degli elementi finiti lineari, utilizzando la function `dirNeusolve` fornita. Si rappresenti in un grafico la soluzione numerica sovrapposta alla soluzione esatta.

Soluzione. Definiamo i dati e rappresentiamo la soluzione esatta:

```
f = @(x) -2*exp(2*x).*(2*x.^2+4*x+1); % forzante
L = 1; % lunghezza intervallo
gN = 4*exp(2); % condizione di Neumann in x = L

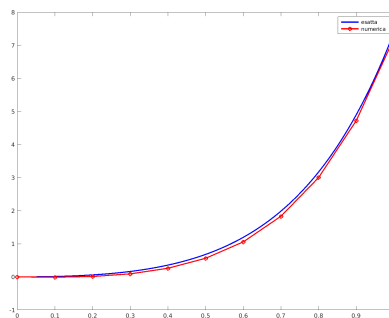
uex = @(x) exp(2*x).*x.^2; % soluzione esatta
```

```
figure(1)
xplot = linspace(0, L, 1000);
plot(xplot, uex(xplot), 'b-', 'linewidth', 2)
hold on
```

Quindi risolviamo il problema numericamente con $h = 1/10$ e sovrapponiamo il grafico ottenuto:

```
[x, u] = dirNeusolve(f, L, gN, L/10);
plot(x, u, 'ro-', 'linewidth', 2)
legend('esatta', 'numerica')
```

Ottenendo il seguente grafico



- (d) (3 punti) [M+T] Si ripeta il calcolo della soluzione numerica per valori di h al variare di h con $h = \frac{1}{N}$, $N = 20, 40, 80$. Calcolare l'errore e rappresentarne l'andamento in scala logaritmica. Commentare il risultato alla luce della teoria. (*Suggerimento*: per il calcolo dell'errore definire l'integranda e utilizzare la funzione `simpcomp` fornita.)

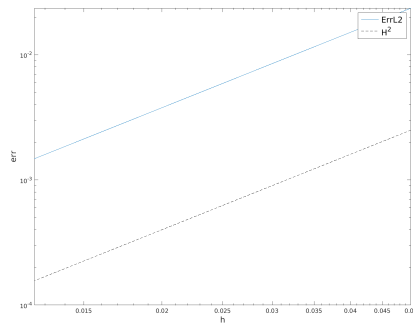
Soluzione. Definiamo un vettore per i passi d. griglia e un vettore, della stessa lunghezza, per il salvataggio dell'errore, quindi impostiamo un ciclo `for`

```
N = [20 40 80];
errL2 = 0*N;
for i = 1:length(N)
    n = N(i);
    h = L/n;
    [x, u] = dirNeusolve(f, L, gN, h);

    integrandaL2 = @(t) (uex(t) - interp1(x, u, t)).^2;
    errL2(i) = sqrt(simpcomp(0, L, 1000, integrandaL2));
end
```

Il calcolo dell'errore è effettuato definendo come funzione integranda il quadrato della differenza fra la soluzione esatta e quella numerica in un generico punto t e integrando numericamente con il metodo di Simpson. Per ottenere il grafico in scala logaritmica utilizziamo i seguenti comandi:

```
figure(2)
H = L./N;
loglog(H, errL2, H, H.^2, 'k--')
legend('ErrL2', 'H^2','FontSize',14)
xlabel('h','FontSize',14)
ylabel('err','FontSize',14)
```



Notiamo che l'errore diminuisce quadraticamente come previsto dalla teoria.