ESERCIZI DA SVOLGERE PER L' ELABORATO
CAP 1
CN A.A. 17/18

1. Sia $x = e \approx 2.7183 = \tilde{x}$. Si calcoli il corrispondente errore relativo ϵ_x e il numero di cifre significative k con cui $\tilde{x}$ approssima x. Si verifichi che
$$|\epsilon_x| \approx \frac{1}{2} 10^{-k} .$$

2. Usando gli sviluppi di Taylor fino al secondo ordine con resto in forma di Lagrange, si verifiche che se $f \in C^3$, risulta
$$f'(x) = \phi_h(x) + O(h^2)$$
dove
$$\phi_h(x) = \frac{f(x+h) - f(x-h)}{2h} .$$

3. Utilizzando Matlab, si costruisca una tabella dove, per $h = 10^{-j}\,, j = 1, \ldots, 10$ e per la funzione $f(x) = x^4$ si riporta il valore di $\phi_h(x)$ definito nell'Esercizio 1 in $x = 1$. Commentare i risultati ottenuti.

4. Si dia una maggiorazione del valore assoluto dell'errore relativo con cui $x + y + z$ viene approssimato dall'approssimazione prodotta dal calcolatore, ossia $(x \oplus y) \oplus z$ (supporre che non ci siano problemi di overflow o di underflow). Ricavare l'analoga maggiorazione anche per $x \oplus (y \oplus z)$ tenendo presente che $x \oplus (y \oplus z) = (y \oplus z) \oplus x$.

5. Eseguire le seguenti istruzioni in Matlab:

$$x = 0;\ count = 0;$$
$$while\ x \sim= 1,\ x = x + delta,\ count = count + 1,\ end$$

dapprima ponendo $delta = 1/16$ e poi ponendo $delta = 1/20$. Commentare i risultati ottenuti e in particolare il non funzionamento nel secondo caso.

6. Verificare che entrambe le seguenti successioni convergono a $\sqrt{3}$, (riportare le successive approssimazioni in una tabella a due colonne, una per ciascuna successione),

$$\begin{aligned} x_{k+1} &= (x_k + \tfrac{3}{x_k})/2\,, & x_0 &= 3; \\ x_{k+1} &= (3 + x_{k-1}x_k)/(x_{k-1} + x_k)\,, & x_0 &= 3;\; x_1 = 2\,. \end{aligned}$$

Per ciascuna delle due successioni, dire quindi dopo quante iterazioni si ottiene un'approssimazione con un errore assoluto minore o uguale a 10^{-12} in valore assoluto.

ESERCIZI DA SVOLGERE PER L' ELABORATO
CAP 2
CN A.A. 17/18

1. Determinare analiticamente gli zeri del polinomio

$$P(x) = x^3 - 4x^2 + 5x - 2$$

e la loro molteplicità. Dire perché il metodo di bisezione è utilizzabile per approssimarne uno a partire dall'intervallo di confidenza $[a\,,\,b] = [0\,,\,3]$. A quale zero di P potrà tendere la successione generata dal metodo di bisezione a partire da tale intervallo? Costruire una tabella in cui si riportano il numero di iterazioni e di valutazioni di P richieste per valori decrescenti della tolleranza `tolx`.

2. Completare la tabella precedente riportando anche il numero di iterazioni e di valutazioni di P richieste dal metodo di Newton, dal metodo delle corde e dal metodo delle secanti (con secondo termine della successione ottenuto con Newton) a partire dal punto $x_0 = 3$. Commentare i risultati riportati in tabella. È possibile utilizzare $x_0 = 5/3$ come punto di innesco?

3. Costruire una seconda tabella analoga alla precedente relativa ai metodi di Newton, di Newton modificato e di accelerazione di Aitken applicati alla funzione polinomiale P a partire dal punto di innesco $x_0 = 0$. Commentare i risultati riportati in tabella.

4. Definire una procedura iterativa basata sul metodo di Newton per approssimare $\sqrt{\alpha}$, per un assegnato $\alpha > 0$. Costruire una tabella dove si riportano le successive approssimazioni ottenute e i corrispondenti errori assoluti (usare l'approssimazione Matlab di $\sqrt{\alpha}$ per il calcolo dell'errore) nel caso in cui $\alpha = 5$ partendo da $x_0 = 5$.

5. Definire una procedura iterativa basata sul metodo delle secanti sempre per approssimare $\sqrt{\alpha}$, per un assegnato $\alpha > 0$. Completare la tabella precedente aggiungendovi i risultati ottenuti con tale procedura partendo da $x_0 = 5$ e $x_1 = 3$. Commentare i risultati riportati in tabella.

ESERCIZI DA SVOLGERE PER L' ELABORATO
CAP 3
CN A.A. 17/18

1. Scrivere una function Matlab per la risoluzione di un sistema lineare con matrice dei coefficienti triangolare inferiore a diagonale unitaria. Inserire un esempio di utilizzo.

2. Utilizzare l'Algoritmo 3.6 del libro per stabilire se le seguenti matrici sono sdp o no,

$$A_1 = \begin{pmatrix} 1 & -1 & 2 & 2 \\ -1 & 5 & -14 & 2 \\ 2 & -14 & 42 & 2 \\ 2 & 2 & 2 & 65 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & -1 & 2 & 2 \\ -1 & 6 & -17 & 3 \\ 2 & -17 & 48 & -16 \\ 2 & 3 & -16 & 4 \end{pmatrix}.$$

3. Scrivere una function Matlab che, avendo in ingresso un vettore $\mathbf{b}$ contenente i termini noti del sistema lineare $A\mathbf{x} = \mathbf{b}$ con A sdp e l'output dell'Algortmo 3.6 del libro (matrice A riscritta nella porzione triangolare inferiore con i fattori L e D della fattorizzazione LDL^T di A), ne calcoli efficientemente la soluzione.

4. Scrivere una function Matlab che, avendo in ingresso un vettore $\mathbf{b}$ contenente i termini noti del sistema lineare $A\mathbf{x} = \mathbf{b}$ e l'output dell'Algortmo 3.7 del libro (matrice A riscritta con la fattorizzazione LU con pivoting parziale e il vettore $\mathbf{p}$ delle permutazioni), ne calcoli efficientemente la soluzione.

5. Inserire alcuni esempi di utilizzo delle due function implementate per i punti 3 e 4, scegliendo per ciascuno di essi un vettore $\hat{\mathbf{x}}$ e ponendo $\mathbf{b} = A\hat{\mathbf{x}}$. Riportare $\hat{\mathbf{x}}$ e la soluzione $\mathbf{x}$ da essi prodotta. Costruire anche una tabella in cui, per ogni esempio considerato, si riportano il numero di condizionamento di A in norma 2 (usare `cond` di Matlab) e le quantità $\|\mathbf{r}\|/\|\mathbf{b}\|$ e $\|\mathbf{x} - \hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}\|$.

6. Sia $A = \begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix}$ con $\epsilon = 10^{-13}$. Definire L triangolare inferiore a diagonale unitaria e U triangolare superiore in modo che il prodotto

LU sia la fattorizzazione LU di A e, posto $\mathbf{b} = A\mathbf{e}$, con $\mathbf{e} = (1,1)^T$, confrontare l'accuratezza della soluzione che si ottiene usando il comando $U\backslash(L\backslash\mathbf{b})$ (Gauss senza pivoting) e il comando $A\backslash\mathbf{b}$ (Gauss con pivoting).

7. Scrivere una function Matlab specifica per la risoluzione di un sistema lineare con matrice dei coefficienti $A \in \mathbb{R}^{n\times n}$ bidiagonale inferiore a diagonale unitaria di Toeplitz, specificabile con uno scalare α. Sperimentarne e commentarne le prestazioni (considerare il numero di condizionamento della matrice) nel caso in cui $n = 12$ e $\alpha = 100$ ponendo dapprima $\mathbf{b} = (1, 101, \cdots, 101)^T$ (soluzione esatta $\hat{\mathbf{x}} = (1, \cdots, 1)^T$) e quindi $\mathbf{b} = 0.1 * (1, 101, \cdots, 101)^T$ (soluzione esatta $\hat{\mathbf{x}} = (0.1, \cdots, 0.1)^T$).

8. Scrivere una function che, dato un sistema lineare sovradeterminato $A\mathbf{x} = \mathbf{b}$, con $A \in \mathbb{R}^{m\times n}, m > n$, $\mathrm{rank}(A) = n$ e $\mathbf{b} \in \mathbb{R}^m$, preso come input $\mathbf{b}$ e l'output dell'Algoritmo 3.8 del libro (matrice A riscritta con la parte significativa di R e la parte significativa dei vettori di Householder normalizzati con prima componente unitaria), ne calcoli efficientemente la soluzione nel senso dei minimi quadrati.

9. Inserire due esempi di utilizzo della function implementata per il punto 8 e confrontare la soluzione ottenuta con quella fornita dal comando $A\backslash\mathbf{b}$.

10. Scrivere una function che realizza il metodo di Newton per un sistema nonlineare (prevedere un numero massimo di iterazioni e utilizzare il criterio di arresto basato sull'incremento in norma eucliedea). Utilizzare la function costruita al punto 4 per la risoluzione del sistema lineare ad ogni iterazione.

11. Verificato che la funzione $f(x_1, x_2) = x_1^2 + x_2^3 - x_1x_2$ ha un punto di minimo relativo in $(1/12, 1/6)$, costruire una tabella in cui si riportano il numero di iterazioni eseguite, e la norma eucliedea dell'ultimo incremento e quella dell' errore con cui viene approssimato il risultato esatto utilizzando la function sviluppata al punto precedente per valori delle tolleranze pari a 10^{-t} , con $t = 3, 6$. Utilizzare $(1/2\,, 1/2)$ come punto di innesco. Verificare che la norma dell'errore è molto più piccola di quella dell'incremento (come mai?)

1 Esercizi Capitolo 4

Esercizio 4.1

Scrivere una function Matlab che implementi il calcolo del polinomio interpolante di grado n in forma di Lagrange. La forma della funcion deve essere del tipo: `y = lagrange( xi, fi, x )`

Esercizio 4.2

Scrivere una function Matlab che implementi il calcolo del polinomio interpolante di grado n in forma di Newton. La forma della funcion deve essere del tipo: `y = newton( xi, fi, x )`

Esercizio 4.3

Scrivere una function Matlab che implementi il calcolo del polinomio interpolante di Hermite. La forma della funcion deve essere del tipo:
`y = hermite( xi, fi, f1i, x )`

Esercizio 4.4

Utilizzare le functions degli esercizi precedenti per disegnare l'approssimazione della funzione $\sin x$ nell'intervallo $[0, 2\pi]$, utilizzando le ascisse di interpolazione $x_i = i\pi$, $i = 0, 1, 2$.

Esercizio 4.5

Scrivere una function Matlab che implementi la *spline* cubica interpolante (naturale o *not-a-knot*, come specificato in ingresso) delle coppie di dati assegnate. La forma della funcion deve essere del tipo:
`y = spline3( xi, fi, x, tipo )`

Esercizio 4.6

Scrivere una function Matlab che implementi il calcolo delle ascisse di Chebyshev per il polinomio interpolante di grado n, su un generico intervallo $[a, b]$. La function deve essere del tipo: `xi = ceby( n, a, b )`.

Esercizio 4.7

Utilizzare le function degli Esercizi 4.1 e 4.6 per graficare l'approssimazione della funzione di Runge sull'intervallo $[-6, 6]$, per $n = 2, 4, 6, , \dots, 40$. Stimare, numericamente, l'errore commesso in funzione del grado n del polinomio interpolante.

Esercizio 4.8

Relativamente al precedente esercizio, stimare numericamente, la crescita della costante di Lebesgue.

Esercizio 4.9

Utilizzare la function dell'Esercizio 4.1 per approssimare la funzione di Runge, sull'intervallo $[-6, 6]$, su una partizione uniforme di $n + 1$ ascisse, $n = 2, 4, 6 \dots, 40$. Stimare le corrispondenti costati di Lebesgue.

Esercizio 4.10

Stimare, nel senso dei minimi quadrati, posizione, velocità iniziale, ed accelerazione, relativo ad un moto rettilineo uniformemente accelerato per cui sono note le seguenti misurazioni delle coppie (tempo,spazio): $(1, 2.9)$, $(1, 3.1)$, $(2, 6.9)$, $(2, 7.1)$, $(3, 12.9)$, $(3, 13.1)$, $(4, 20.9)$, $(4, 21.1)$, $(5, 30.9)$, $(5, 31.1)$.

2 Esercizi Capitolo 5

Esercizio 5.1

Scrivere una function Matlab che implementi la formula composita dei trapezi su $n + 1$ ascisse equidistanti nell'intervallo $[a, b]$, relativamente alla funzione implementata da `fun(x)`. La function deve essere del tipo:

```
If = trapcomp( n, a, b, fun )
```

Esercizio 5.2

Scrivere una function Matlab che implementi la formula composita di Simpson su $2n + 1$ ascisse equidistanti nell'intervallo $[a, b]$, relativamente alla fun-

zione implementata da `fun(x)`. La function deve essere del tipo:
`If = simpcomp( n, a, b, fun )`

Esercizio 5.3

Scrivere una function Matlab che implementi la formula composita dei trapezi adattativa nell'intervallo $[a, b]$, relativamente alla funzione implementata da `fun(x)`, e con tolleranza `tol`. La function deve essere del tipo:
`If = trapad( a, b, fun, tol )`

Esercizio 5.4

Scrivere una function Matlab che implementi la formula composita di Simpson adattativa nell'intervallo $[a, b]$, relativamente alla funzione implementata da `fun(x)`, e con tolleranza `tol`. La function deve essere del tipo:
`If = simpad( a, b, fun, tol )`

Esercizio 5.5

Calcolare quante valutazioni di funzione sono necessarie per ottenere una approssimazione di

$$\mathcal{I}(f) = \int_0^1 \exp(-10^6 x)\,\mathrm{d}x,$$

che vale 10^{-6}, in doppia precisione IEEE, con una tolleranza 10^{-9}, utilizzando le functions dei precedenti esercizi. Argomentare quantitativamente la risposta.

3 Esercizi capitolo 6

Esercizio 6.1

Scrivere una function Matlab che generi la matrice *sparsa* $n \times n$, con $n > 10$,

$$A_n = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}, \quad \text{con} \quad a_{ij} = \begin{cases} 4, & \text{se} \quad i = j, \\ -1, & \text{se} \quad i = j \pm 1, \\ -1, & \text{se} \quad i = j \pm 10. \end{cases} \qquad (1)$$

Utilizzare, a questo fine, la function Matlab `spdiags`.

Esercizio 6.2

Utilizzare il metodo delle potenze per calcolarne l'autovalore dominante della matrice A_n del precedente esercizio, con una approssimazione $tol = 10^{-5}$, partendo da un vettore con elementi costanti. Riempire, quindi, la seguente tabella:

n	numero di iterazioni effettuate	stima autovalore
100		
200		
$\vdots$		
1000		

Esercizio 6.3

Utilizzare il metodo di Jacobi per risolvere il sistema lineare

$$A_n\mathbf{x} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \tag{2}$$

dove A_n è la matrice definita in (1), con tolleranza $tol = 10^{-5}$, e partendo dal vettore nullo. Graficare il numero di iterazioni necessarie, rispetto alla dimensione n del problema, con n che varia da 100 a 1000 (con passo 20).

Esercizio 6.4

Ripetere una procedura analoga a quella del precedente esercizio utilizzando il medodo di Gauss-Seidel.

Esercizio 6.5

Con riferimento al sistema lineare (2), con $n = 1000$, graficare la norma dei residui, rispetto all'indice di iterazione, gererati dai metodi di Jacobi e Gauss-Seidel. Utilizzare il formato `semilogy` per realizzare il grafico, corredandolo di opportune *label*.

Nota bene.

Inserire, nell'elaborato, i codici utilizzati per svolgere gli Esercizi 6.2–6.5.