Serie 1
Introduzione a Matlab®

1 Vettori, Matrici, Funzioni e Grafici di Funzione

Esercizio 1.1

Costruire i seguenti vettori

$$\begin{aligned}
v_1 &= [1, 2, 4, \ldots, 1024], \\
v_2 &= [\cos(\pi), \cos(\pi/2), \ldots, \cos(\pi/10)]^T, \\
v_3 &= [0.1, 0.05, 0.025, \ldots, 0.003125], \\
v_4 &= [e+6, 0, e^2-11, 0, e^3+16, 0, \ldots, e^9+46, 0, e^{10}-51].
\end{aligned}$$

Esercizio 1.2

Costruire la matrice

$$A = \begin{bmatrix} 2 & 0 & 10 & 0 & 40 \\ 5 & 2 & 0 & 10 & 0 \\ 10 & 5 & 2 & 0 & 10 \\ 0 & 10 & 5 & 2 & 0 \\ 40 & 0 & 10 & 5 & 2 \end{bmatrix}$$

Sommare tutti gli elementi della matrice A ed estrarre le seguenti sottomatrici:

$$A_1 = \begin{bmatrix} 2 & 0 & 10 \\ 5 & 2 & 0 \\ 10 & 5 & 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 2 & 0 & 0 \\ 10 & 5 & 0 \\ 40 & 0 & 5 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 5 & 0 & 10 \\ 10 & 2 & 0 \\ 0 & 5 & 2 \end{bmatrix}.$$

Esercizio 1.3

Costruire le seguenti matrici

$$B = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 & 1 & 0 \\ 1 & 1 & & & & & 1 \\ 1 & & 1 & & 0 & & 1 \\ \vdots & & & \ddots & & & \vdots \\ 1 & & 0 & & 1 & & 1 \\ 1 & & & & & 1 & 1 \\ 0 & 1 & 1 & \ldots & 1 & 1 & 1 \end{bmatrix} \in \mathbb{R}^{10\times 10},$$

$$C = \begin{bmatrix} 1 & 1 & 0.5 & & & & \\ 1 & 2 & 1 & 0.5 & & 0 & \\ 0.5 & 1 & 3 & 1 & 0.5 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0.5 & 1 & 198 & 1 & 0.5 \\ & 0 & & 0.5 & 1 & 199 & 1 \\ & & & & 0.5 & 1 & 200 \end{bmatrix} \in \mathbb{R}^{200\times 200},$$

$$D = \begin{bmatrix} 20 & 1 & 0 & & & & \\ 0 & 18 & 3 & 0 & & 0 & \\ 0.5 & 0 & 16 & 9 & 0 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0.5 & 0 & 6 & 2187 & 0 \\ & 0 & & 0.5 & 0 & 4 & 6561 \\ & & & & 0.5 & 0 & 2 \end{bmatrix} \in \mathbb{R}^{10\times 10},$$

[**Sfida!** È possibile costruire B con 3 istruzioni... 2 istruzioni... o con 1 sola istruzione!!!]

Esercizio 1.4

Valutare, tramite i comandi Matlab® `eval`, `inline` e *anonymous function*, le seguenti funzioni:

- $f(x) = x\sin(x) + \left(\frac{1}{2}\right)^{\sqrt{x}}$
- $f(x) = x^4 + \log(x^3+1)$

sul vettore `x=[0 1 2 3]` .

Esercizio 1.5

Disegnare il grafico della funzione

$$f(x) = 2 + (x-3)\sin(5(x-3))$$

per $0 \le x \le 6$ (vedi figura 1). Sovrapporre a questo grafico quelli delle due rette che limitano l'andamento di tale funzione, disegnate con linea tratteggiata.

Suggerimento: le due rette in questione sono $y = -x+5$ e $y = x-1$.

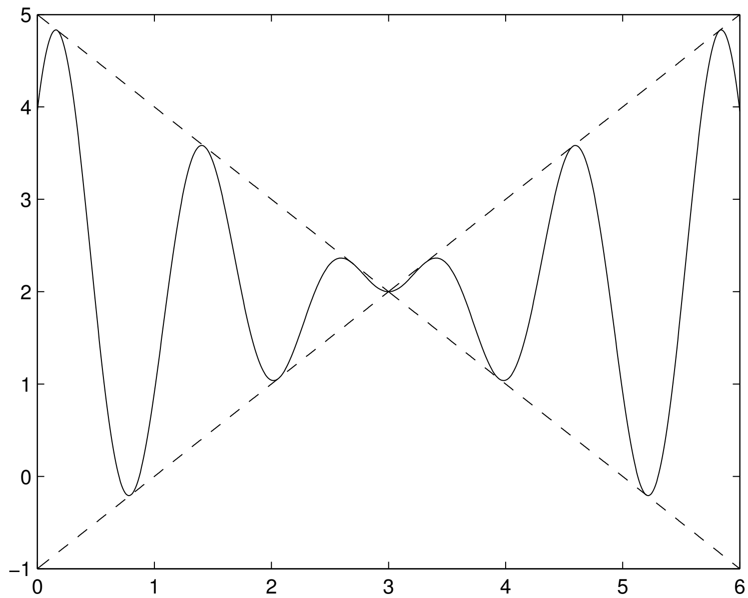

Figura 1: Soluzione dell'esercizio 1.5

Esercizio 1.6

Cosa ci si aspetta di ottenere disegnando il grafico x-logaritmico di $f(x) = (\log(x))^2$ sull'intervallo $0.1 \le x \le 10$? Disegnare il grafico e verificare.

2 Aritmetica Finita del Calcolatore

È importante notare che l'insieme dei numeri utilizzabili da un calcolatore è un particolare sottoinsieme finito dei numeri razionali che chiamiamo $\mathbb{F}$.

Per memorizzare ogni numero è disponibile una quantita limitata di bit, più precisamente 64; di conseguenza i numeri che costituiscono l'aritmetica del calcolatore hanno necessariamente un numero finto di cifre decimali. In particolare, i numeri irrazionali come π oppure i numeri razionali periodici come $1/3$ sono sostituiti da una loro approssimazione con un numero finito di cifre decimali.

Se in $\mathbb{F}$ sono contenuti solo numeri con un numero finito di cifre decimali, attorno ad ogni elemento f di $\mathbb{F}$ esiste un piccolo intervallo I_f "vuoto", che contiene solo f stesso e nessun altro elemento di $\mathbb{F}$.

In altre parole, la distanza fra f e il suo elemento successivo di $\mathbb{F}$ non è infinitamente piccola, ma è un valore ben determinato (per quanto "molto piccolo"). Tale distanza si definisce *epsilon macchina*, e si indica tramite $\epsilon(f)$.

In Matlab® (e Octave) esiste la variabile predefinita `eps` che rappresenta l'epsilon macchina del numero `1`, cioè il più piccolo numero che sommato al numero `1` fornisce un numero maggiore di `1`. In particolare:

```
>> eps
```

```
ans =

   2.2204e-16
```

Memorizziamo in una variabile `a` il valore `1 + eps`, osservando che anche con il formato di visualizzazione `format long e` la variabile `a` sembra ancora essere uguale al numero `1`.

```
 >> format long e
>> a = 1+eps

a =

     1.000000000000000e+00
```

Proviamo ora a sottrarre ad `a` il numero `1` e riotteniamo esattamente il valore di `eps`:

```
 >> a - 1

ans =

     2.220446049250313e-16
```

Adesso eseguiamo lo stesso procedimento sommando al numero `1` il valore `eps/2` e otteniamo:

```
 >> format long e
>> b = 1+eps/2

b =

     1
```

Apparentemente non c'è nessuna differenza con il caso precedente, ma se adesso sottraiamo `1` da `b` otteniamo:

```
 >> b - 1

ans =

      0
```

Come è stato accenato in precedenza, l'epsilon macchina non è lo stesso per ogni numero di $\mathbb{F}$ e dipende dalla tecnica usata per gestire i 64 bit disponibili per la rappresentazione di ogni numero. In particolare in Matlab® esiste la funzione `eps( )` che prende in ingresso un numero e restituisce il corrispondente epsilon macchina. Per esempio:

comando	risultato
» eps(1)	2.2204e-16
» eps(10)	1.7764e-15
» eps(100)	1.4211e-14
» eps(1000)	1.1369e-13
...	...

Possiamo osservare che `eps(1)` coincide con il valore della variabile predefinita `eps`. In particolare è molto importante notare che all'aumentare del numero in considerazione, il corrispondente epsilon macchina aumenta. Questo significa che i numeri che costituiscono l'insieme $\mathbb{F}$ non sono equispaziati, ma sono piu addensati intorno ai numeri piccoli e si diradano man mano che il loro valore aumenta. Proviamo questa affermazione con un esempio:

```
 >> format long
 >> 1000 + eps(1) -1000

ans =

     0

>> 1 + eps(1) - 1

ans =

     2.220446049250313e-16
```

da cui deduciamo che il calcolatore non è in grado di interpretare il numero `eps(1)=2.2204e-16` come un incremento diverso da zero quando viene sommato al numero `1000`, mentre lo riconosce come numero non nullo quando viene sommato ad `1`. Il più piccolo incremento del numero `1000` riconosciuto dal calcolatore è il corrispondente epsilon macchina:

```
>> 1000 + eps(1000) - 1000

ans =

     1.136868377216160e-13
```

Osserviamo infine che la funzione built-in `eps( )` di Matlab® di cui abbiamo parlato, non è implementata in Octave mentre la variabile predefinita `eps` è la stessa. Infatti in Octave è implementata come built-in un'altra funzione `eps( )`:

```
 octave:> n = 2
 octave:> eps(n)
ans =

   2.2204e-16   2.2204e-16
   2.2204e-16   2.2204e-16
```

produce una matrice quadrata di dimensione `n` di valori tutti coincidenti con la variabile `eps`.

Come ultima considerazione, possiamo identificare l'elemento più piccolo e più grande di $\mathbb{F}$ con le variabili predefinite `realmin` e `realmax` di Matlab® e Octave :

```
>> realmin

ans =
```

```
    2.225073858507201e-308

>> realmax

ans =

    1.797693134862316e+308
```

Assegnando ad una variabile un numero maggiore di `realmax`, Matlab® non è in grado di interpretare l'istruzione di assegnazione:

```
 >> a = 1e400

a =

   Inf
```

ma non viene prodotto un messaggio di errore. Differentemente, è ancora possibile assegnare ad una variabile un valore inferiore di `realmin` e Matlab® è ancora in grado di interpretare correttamente l'istruzione:

```
>> a = 1e-310

a =

    9.999999999999969e-311
```

Il più piccolo numero in assoluto riconoscibile da Matlab® è l'epsilon macchina di `realmin`:

```
 >> eps(realmin)

ans =

    4.940656458412465e-324
```

e assegnando ad una variabile inferiore ad `eps(realmin)` Matlab® interpreta l'istruzione come assegnazione di un valore nullo:

```
 >> a = 1e-325

a =

     0
```

3 Esempi di Errori Numerici

Vediamo alcuni esempi di errori numerici che ci permettono di comprendere meglio che tipo di problemi sono legati all'aritmetica finita del calcolatore. In base alle conoscenze acquisite fino ad ora, è possibile definire un vettore di elementi equispaziati "a mano" oppure con il comando `linspace`:

```
>> a = [0:0.01:1];
>> b=linspace(0,1,101);
>> whos
  Name       Size                Bytes  Class     Attributes

  a          1x101                 808  double
  b          1x101                 808  double
```

Osserviamo che i due vettori `a` e `b` sono costituiti dallo stesso numero di elementi e anche gli estremi del loro intervallo di definizione `[0,1]` sono gli stessi. Proviamo a fare un grafico della differenza, in valore assoluto tra ogni elemento di posto corrispondente dei due vettori. Definiamo il vettore

```
>> c = abs( a-b );
>> plot(c, '*-')
```

e osserviamo il grafico del vettore `c` in figura 2.

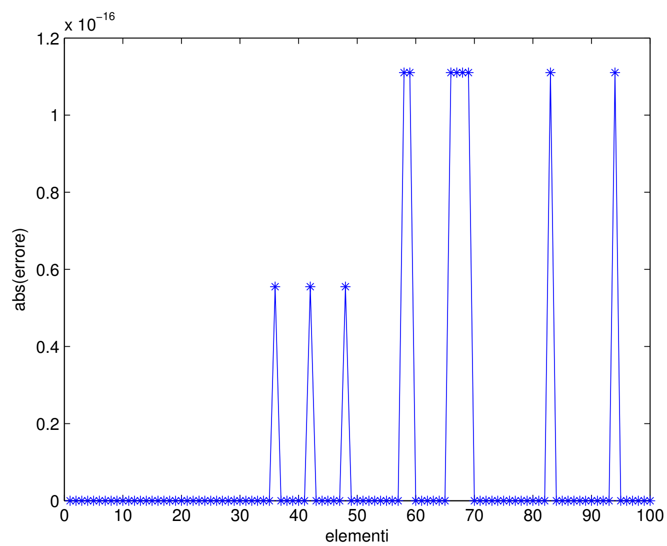

Figura 2: Grafico del vettore c

Si può notare che parecchi elementi sono affetti da errore e in particolare l'errore aumenta verso l'estremo di destra dell'intervallo.

Un altro esempio è il fenomeno noto come *cancellazione di cifre significative*. Calcolando con Matlab® l'espressione $((x + 1) - 1)/x$ per $x = 10^{-15}$, il cui risultato è ovviamente 1, si ottiene:

```
>> x=1e-15
```

```
x =
   1.0000e-15
>> ((1+x)-1)/x
ans =
    1.1102
```

Il motivo di questo fenomeno è che la somma fra numeri di $\mathbb{F}$ in valore assoluto molto diversi non è precisa, perchè nelle conversioni che il calcolatore esegue per sommare i due numeri potrebbero perdersi alcune cifre significative. Infatti:

```
>> (1+x)-1
ans =
   1.1102e-15
```

mentre ovviamente:

```
>> (1-1)+x
ans =
   1.0000e-15
```

quindi possiamo concludere che in generale la somma non è associativa in $\mathbb{F}$. In generale è meglio evitare di sommare numeri molto diversi fra loro, e se è necessario farlo cercare perlomeno di riordinare gli addendi dal più grande al più piccolo.

4 Cicli, Script, Funzioni, Outputs

Esercizio 4.1

1. Scrivere uno script `mat_hilbert.m` che fornisca la matrice di Hilbert avente dimensione 5×5, i cui elementi $a_{i\,j}$ sono definiti da:

$$a_{i\,j} = \frac{1}{i + j - 1}\,.$$

   (SUGGERIMENTO: utilizzare due cicli for annidati.)

2. Verificare il corretto funzionamento della funzione del punto precedente, confrontandone il risultato con la matrice restituita dal comando `hilb(5)`.

Esercizio 4.2

Una famoso problema non ancora risolto è quello della convergenza della successione definita come segue:

dato $n_0 \in \mathbb{N}$, $n_0 > 1$, $\forall\ i = 1, 2, \ldots$

$$n_{i+1} = \begin{cases} n_i/2 & \text{se } n_i \text{ è pari,} \\ 3n_i + 1 & \text{se } n_i \text{ è dispari e } n_i \neq 1. \end{cases}$$

Quando $n_i = 1$ la successione termina.

Per esempio, se $n_0 = 7$ la successione è la seguente: 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. Chiaramente la successione converge logaritmicamente in n_0 non appena sia $n_0 = 2^N$, con N intero positivo o nullo.

Scrivere uno script `treuno.m` che permetta di scegliere n_0 arbitrario e memorizzi la successione in un vettore. (SUGGERIMENTO: si può utilizzare la funzione `mod`. Per maggiori informazioni: `help mod`)

Esercizio 4.3

Utilizzare un ciclo `while` per determinare quanti anni occorrono per accumulare un milione di euro in un conto corrente bancario se vengono depositati 10 mila euro iniziali, 10 mila alla fine di ogni anno e se la banca riconosce un interesse annuo del 2% sui conti correnti.

Esercizio 4.4

Per il calcolo di π si può utilizzare il seguente metodo: si generano n coppie $\{(x_k, y_k)\}$ di numeri casuali compresi tra 0 e 1 e di questi si calcola il numero m di punti che cadono nel primo quarto del cerchio di centro l'origine e raggio 1. Si ha che π è il limite per $n \to \infty$ dei rapporti $\pi_n = 4m/n$. Si scriva un programma che esegua questo calcolo e si verifichi la correttezza del risultato al crescere di n. (SUGGERIMENTO: utilizzare la funzione `rand`.)

Esercizio 4.5

Determinare il primo numero intero n tale che $\displaystyle\sum_{k=1}^{n} k >= 88$, dove $\displaystyle\sum_{k=1}^{n} k = 1 + 2 + 3 + .. + n$. Si utilizzi un ciclo `while` per controllare la somma al crescere di k.

Esercizio 4.6

Definire la colonna di valori a di lunghezza $n = 10$ tale che:

$$a_k = \begin{cases} \dfrac{1}{k} & \text{se } k = 2, \text{ oppure } k = 6 \\ \dfrac{1}{(k-2)(k-6)} & \text{altrimenti} \end{cases}$$

Si utilizzi il ciclo `for` per definire ogni elemento della lista e l'istruzione `if` per controllare l'indice di tale elemento.

Esercizio 4.7

Si definisca la riga di valori $$\underline{v}_k = [1\,, 9\,, 25\,, \ldots, (2k+1)^2]$$

con $k = 8$ usando un ciclo `for` per definire ciascun elemento uno alla volta.

Esercizio 4.8

Si definisca una funzione che, dato come input il valore k, ritorni il corrispondente vettore $\underline{v}_k$ come definito nell'esercizio precedente.

Esercizio 4.9

Con l'algoritmo di Erone si può determinare la radice quadrata di un numero reale positivo n, come il limite per $k \to \infty$ della successione

$$r_k = \begin{cases} n & \text{se } k = 0 \\ \dfrac{1}{2}\left(r_{k-1} + \dfrac{n}{r_{k-1}}\right) & \text{se } k = 1, 2, 3 \ldots \end{cases}$$

Scrivere una funzione che prende in ingresso n e mediante l'algoritmo di Erone con $toll = 1e{-}6$ restituisce un'approssimazione della radice quadrata di n e il relativo grafico della successione r_k generata.

Esercizio 4.10

Approssimare π utilizzando l'algoritmo di Archimede:

$$p_n = 2^n a_n \quad \text{con} \quad a_n = \begin{cases} \sqrt{2} & \text{se } n = 1 \\ \sqrt{2}\sqrt{1 - \sqrt{1 - \dfrac{1}{4}a_{n-1}^2}} & \text{se } n = 2, 3, \ldots \end{cases}$$

$$q_n = 2^n b_n \quad \text{con} \quad b_n = \begin{cases} 2 & \text{se } n = 1 \\ \dfrac{a_n}{\sqrt{1 - \frac{1}{4}a_{n-1}^2}} & \text{se } n = 2, 3, \ldots \end{cases}$$

dove p_n è monotona crescente mentre q_n è monotona decrescente e entrambe approssimano π rispettivamente per difetto ed eccesso.

Si implementi il calcolo degli elementi delle successioni e si mostri graficamente che esse tendono a π.

Esercizio 4.11

1. Rappresentare graficamente la circonferenza di equazione $x^2 + y^2 = 1$.
2. Rappresentare graficamente l'ellisse di equazione $\dfrac{x^2}{4} + \dfrac{y^2}{9} = 1$.
3. Rappresentare graficamente il quadrato di lato 5 centrato nell'origine con i lati paralleli agli assi cartesiani.

Esercizio 4.12

Rapprersentare la seguente funzione definita a tratti:

$$f(x) = \begin{cases} -\sqrt{(x^2 - x)} & x < 0 \\ (-x^2 + 2x)e^{-x} & x \geq 0. \end{cases}$$

Esercizio 4.13

Scrivere una funzione che restituisce una tabella quadrata tale che su ogni riga i e colonna j ci sia il valore $2 * i * j + (\mathcal{K} + 1)$ dove $\mathcal{K}$ è la penultima cifra del vostro numero di matricola.

Esercizio 4.14

Porre $x = 3$, $y = -2$ e valutare con MATLAB la seguente espressione

$$\arctan\left(\frac{y}{x}\right) - \left(\sin\left(x\sqrt{|y|}\right)\right)^2.$$

Esercizio 4.15

Implementare la successione definita per ricorrenza:

$$\begin{cases} a_1 = 0.5 \\ a_n = r a_{n-1}\left(1 - a_{n-1}\right) \quad \text{per } n \geq 2 \end{cases}$$

dove $r = 2.5 + mod(\mathcal{K}, 2)$ e $\mathcal{K}$ è la penultima cifra del vostro numero di matricola.
Rappresentare i primi 100 termini di tale sucessione e scrivere, se esiste, il primo valore a_n (con $n \leq 100$) tale che $|a_n - a_{n-1}| < 1e - 3$.

Esercizio 4.16

Scrivere una funzione che presi tre valori a, b, k restituisce $(a + b)^n$, dove $n = mod(\mathcal{K}, 4) + 3$, utilizzando la formula:

$$(a + b)^n = \sum_{j=0}^{n} \frac{n!}{j!(n - j)!} a^j b^{n-j} :, ,$$

dove $\mathcal{K}$ è l'ultima cifra del vostro numero di matricola.

Esercizio 4.17

Scrivere una funzione che preso in ingresso un vettore con elementi tutti distinti tra loro, ritorni il vettore ordinato in ordine decrescente (non utilizzare l'apposito comando matlab `sort`).

Esercizio 4.18

Calcolare la somma S_p dei numeri pari x e la somma S_d dei numeri dispari y tali che $(5 * \mathcal{K} + \mathcal{K}) \leq x, y \leq 1000$.

Esercizio 4.19

Rappresentare la funzione f utilizzando colori differenti nei due intervalli indicati: $f(x) = \begin{cases} \sin(x) & \text{se } -\pi \leq x \leq 0 \\ \dfrac{x^2+x}{6} & \text{se } 0 \leq x \leq \pi \end{cases}$

Esercizio 4.20

Scrivere una funzione che, dato in ingresso un numero intero n, ritorni se n è un quadrato perfetto (usare il comando disp e floor).

Esercizio 4.21

Calcolare la somma dei numeri naturali $x, 0 \leq x \leq 1000$, tali che $\mathcal{K}+2$ non divide x. Definire $\mathcal{K}$ come l'ultima cifra del vostro numero di matricola.

Esercizio 4.22

Scrivere una funzione che prese le lunghezze dei lati a, b, c di un triangolo restituisce se esso è un triangolo rettangolo.
Utilizzare Teorema di Pitagora e il comando disp.

Esercizio 4.23

Scrivere una funzione che, dato in ingresso un valore `n`, ritorni la matrice di `n`$\times$`n` valori $T = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots \\ 0 & 1 & 0 & 1 & \dots \\ 1 & 0 & 1 & 0 & \dots \\ 0 & 1 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$.

Esercizio 4.24

Scrivere una funzione che, data in ingresso una lista a di n valori $a_i,\ i = 1, 2, \dots, n$, calcoli la media armonica di questi ultimi, definita come $\overline{a} = \left(\dfrac{1}{n} \sum_{i=1}^{n} a_i^{-1} \right)^{-1} = \dfrac{n}{\frac{1}{a_1} + \frac{1}{a_2} + \ \dots + \frac{1}{a_n}}$.

Applicare poi la funzione alla lista di numeri $a = [1, 2, 3, \dots, 10 + \mathcal{K}]$ e riportare il risultato ottenuto in formato `short`.

Esercizio 4.25

Scrivere una funzione che, data in ingresso una lista a di n valori $a_i,\ i = 1, 2, \dots, n$, calcoli la media geometrica di questi ultimi, definita come $\overline{a} = \sqrt[n]{a_1\, a_2\, a_3 \dots a_n}$.
Applicare poi la funzione alla lista di numeri $a = [1, 2, 3, \dots, 10 + \mathcal{K}]$ e riportare il risultato ottenuto in formato `short`.

Esercizio 4.26

Mostrare che la successione ricorsiva definita da $\begin{cases} a_0 = 1 \\ a_n = \dfrac{(a_{n-1})^2 + 2}{2a_{n-1}} & n = 1, 2, \dots \end{cases}$ converge a $\sqrt{2}$ riportando su un grafico l'evoluzione dei valori della successione.

Esercizio 4.27

Scrivere una funzione che, dato in ingresso un valore `n`, ritorni la matrice di `n`×`n` valori
$$T = \begin{bmatrix} 1 & 2 & 3 & \dots & n \\ 0 & 1 & 2 & \dots & n-1 \\ 0 & 0 & 1 & \dots & n-2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Esercizio 4.28

Disegnare il grafico della funzione $f(x) = \dfrac{x^2 + 3}{x - 1}$ nell'intervallo $[-5, 5]$.