



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

Relazione relativa al progetto d'esame di Programmazione Logica e Funzionale

sessione estiva — a.a. 2023/2024

Corso di Laurea in Informatica Applicata
Università di Urbino

STUDENTI:

Barzotti Nicolas
matricola: 313687
Ramagnano Gabriele
matricola: 315439

DOCENTE:

Marco Bernardo

Indice

1 Specifica del Problema	2
2 Analisi del Problema	3
2.1 Dati di Ingresso del Problema	3
2.2 Dati di Uscita del Problema	3
2.3 Relazioni Intercorrenti tra i Dati del Problema	3
3 Progettazione dell'Algoritmo	7
3.1 Scelte di Progetto	7
3.2 Passi dell'Algoritmo	7
4 Implementazione dell'algoritmo	10
4.1 Haskell	10
4.2 Prolog	20
5 Testing	31
5.1 Haskell	31
5.2 Prolog	45

1 Specifica del Problema

Scrivere un programma Haskell e un programma Prolog che, per ogni simulazione numerica da effettuare, acquisiscono da tastiera un insieme finito di parametri numerici e poi stampano su schermo il risultato del calcolo numerico. Per simulazione numerica intendiamo la simulazione di un processo fisico mediante calcolatore, ove con simulazione di un processo fisico si fa riferimento alla rappresentazione, eventualmente approssimata, di tale processo mediante la risoluzione di equazioni e modelli matematici al calcolatore¹. Le simulazioni numeriche coinvolte sono quattro: integrazione di equazioni differenziali di moto fugoide senza attrito, di moto fugoide con attrito, di convezione lineare a una dimensione ed di Burgers a una dimensione.

¹[https://www.treccani.it/enciclopedia/simulazioni-di-processi-fisici-mediante-calcolatore_\(Encyclopaedia-della-Scienza-e-della-Tecnica\)/](https://www.treccani.it/enciclopedia/simulazioni-di-processi-fisici-mediante-calcolatore_(Encyclopaedia-della-Scienza-e-della-Tecnica)/)

2 Analisi del Problema

2.1 Dati di Ingresso del Problema

I dati in ingresso al problema sono stati suddivisi in base all'equazione da integrare numericamente, ne segue quindi la loro descrizione.

Fugoide Senza Attrito

Il dato in ingresso è un numero reale maggiore di zero. Questo rappresenta il passo temporale per l'integrazione dell'equazione del moto fugoide senza attrito.

Fugoide Con Attrito

Il dato in ingresso è un numero reale maggiore di zero. Questo rappresenta il passo temporale per l'integrazione dell'equazione del moto fugoide con attrito.

Convezione

I dati in ingresso per l'integrazione dell'equazione di convezione sono:

1. un numero naturale, rappresenta il numero di punti della funzione d'onda;
2. un numero reale maggiore di zero, rappresenta la lunghezza del passo temporale.

Burgers

Il dato in ingresso per l'integrazione dell'equazione di Burgers è un numero naturale. Questo rappresenta il numero di punti della funzione d'onda.

2.2 Dati di Uscita del Problema

Fugoide Senza Attrito

Il dato in uscita dell'integrazione dell'equazione del moto fugoide senza attrito è una sequenza di numeri reali che rappresentano la funzione di traiettoria dell'areomobile.

Fugoide Con Attrito

Il dato in uscita dell'integrazione dell'equazione del moto fugoide con attrito è una sequenza di numeri reali che rappresentano la funzione di traiettoria dell'areomobile.

Convezione

Il dato in uscita dell'integrazione dell'equazione di convezione lineare a una dimensione è una sequenza di numeri reali che rappresentano i valori finali della funzione d'onda quadra.

Burgers

Il dato in uscita all'integrazione dell'equazione di Burgers a una dimensione è una sequenza di numeri reali che rappresentano i valori finali della funzione a dente di sega.

2.3 Relazioni Intercorrenti tra i Dati del Problema

Fugoide Senza Attrito

L'equazione per il moto di fugoide senza attrito è un'equazione differenziale ordinaria del secondo ordine:

$$z(t)'' = g - \frac{g z(t)}{z_t} = g \left(1 - \frac{z(t)}{z_t}\right). \quad (1)$$

Possiamo trasformare questa equazione del secondo ordine in un sistema di equazioni del primo ordine:

$$z'(t) = b(t) \quad (2)$$

$$b'(t) = g \left(1 - \frac{z(t)}{z_t} \right). \quad (3)$$

Un altro modo di considerare un sistema di due equazioni ordinarie del primo ordine è scrivere il sistema differenziale come un'unica equazione vettoriale:

$$\vec{u} = \begin{pmatrix} z \\ b \end{pmatrix} \quad (4)$$

$$\vec{u}'(t) = \begin{pmatrix} b \\ g - g \frac{z(t)}{z_t} \end{pmatrix}. \quad (5)$$

La soluzione approssimativa al tempo t_n è u_n e la soluzione numerica dell'equazione differenziale consiste nel calcolare una sequenza di soluzioni con la seguente equazione:

$$u_{n+1} = u_n + \Delta t f(u_n) + O(\Delta t)^2. \quad (6)$$

Questa formula è chiamata metodo di Eulero. Per le equazioni di moto fugoide, il metodo di Eulero fornisce il seguente algoritmo:

$$z_{n+1} = z_n + \Delta t b_n \quad (7)$$

$$b_{n+1} = b_n + \Delta t \left(g - \frac{g}{z_t} z_n \right) \quad (8)$$

dove:

- Δt è la lunghezza del passo temporale;
- g è la forza gravitazionale terrestre;
- z_n è l'altitudine del velivolo al passo n ;
- z_t è l'altitudine centrale del velivolo;
- b_n è la velocità del velivolo al passo n .

Il numero di passi di simulazione n viene calcolato $n = \frac{T}{\Delta t}$, dove T è il tempo totale di simulazione. La condizione iniziale è il valore della derivata al tempo $t = 0$ ed è l'altitudine iniziale del velivolo z_0 .

Fugoide Con Attrito

L'equazione per il moto fugoide con attrito è un sistema di equazioni differenziali ordinarie del primo ordine:

$$m \frac{dv}{dt} = -W \sin \theta - D \quad (9)$$

$$mv \frac{d\theta}{dt} = -W \cos \theta + L. \quad (10)$$

Per visualizzare le traiettorie di volo previste da questo modello, che dipendono sia dalla velocità di avanzamento v sia dall'angolo della traiettoria θ , occorre calcolare la posizione dell'aliante ad ogni istante di tempo t . La posizione dell'aliante su un piano verticale sarà determinata dalle coordinate (x, y) :

$$x'(t) = v \cos(\theta) \quad (11)$$

$$y'(t) = v \sin(\theta). \quad (12)$$

L'intero sistema di equazioni discretizzate con il metodo di Eulero è:

$$v^{n+1} = v^n + \Delta t \left(-g \sin \theta^n - \frac{C_D}{C_L} \frac{g}{v_t^2} (v^n)^2 \right) \quad (13)$$

$$\theta^{n+1} = \theta^n + \Delta t \left(-\frac{g}{v^n} \cos \theta^n + \frac{g}{v_t^2} v^n \right) \quad (14)$$

$$x^{n+1} = x^n + \Delta t v^n \cos \theta^n \quad (15)$$

$$y^{n+1} = y^n + \Delta t v^n \sin \theta^n. \quad (16)$$

Scritto in forma vettoriale risulta:

$$u'(t) = f(u)$$

$$u = \begin{pmatrix} v \\ \theta \\ x \\ y \end{pmatrix} \quad f(u) = \begin{pmatrix} -g \sin \theta - \frac{C_D}{C_L} \frac{g}{v_t^2} v^2 \\ -\frac{g}{v} \cos \theta + \frac{g}{v_t^2} v \\ v \cos \theta \\ v \sin \theta \end{pmatrix} \quad (17)$$

dove:

- Δt è la lunghezza del passo temporale;
- g è la forza gravitazionale terrestre;
- x è lo spostamento orizzontale del velivolo;
- y è l'altitudine del velivolo;
- v è la velocità del velivolo;
- v_t è la velocità di trim;
- θ è l'angolo d'inclinazione del velivolo;
- C_D è il coefficiente di resistenza dell'aria;
- C_L è il coefficiente di portanza.

Il numero di passi di simulazione n viene calcolato $n = \frac{T}{\Delta t}$, dove T è il tempo totale di simulazione. Le condizioni iniziali sono rappresentate dalle costanti di integrazione definite dal valore della derivata al tempo $t = 0$:

$$\begin{aligned} v(0) &= v_0 \quad \text{and} \quad \theta(0) = \theta_0 \\ x(0) &= x_0 \quad \text{and} \quad y(0) = y_0. \end{aligned}$$

Convezione

L'equazione di convezione lineare unidimensionale è un'equazione differenziale alle derivate parziali:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0. \quad (18)$$

Per la soluzione numerica di $u(x, t)$ si sono utilizzati i pedici per denotare la posizione spaziale, come in u_i , e gli apici per denotare l'istante temporale, come in u^n :

$$\begin{array}{ccc} \bullet & \bullet & \bullet \\ u_{i-1}^{n+1} & u_i^{n+1} & u_{i+1}^{n+1} \\ \bullet & \bullet & \bullet \\ u_{i-1}^n & u_i^n & u_{i+1}^n \\ \bullet & \bullet & \bullet \\ u_{i-1}^{n-1} & u_i^{n-1} & u_{i+1}^{n-1}. \end{array}$$

L'equazione per fornire la soluzione numerica del problema è data da:

$$u_i^{n+1} = u_i^n - c \frac{\Delta t}{\Delta x} (u_i^n - u_{i-1}^n) \quad (19)$$

dove:

- Δt è la lunghezza del passo temporale;
- Δx è la lunghezza del passo spaziale;
- c è la velocità dell'onda.

Le condizioni iniziali per una funzione d'onda quadra sono definite così:

$$u(x, 0) = \begin{cases} 2 & \text{dove } 0.5 \leq x \leq 1, \\ 1 & \text{altrimenti} \end{cases} \quad (20)$$

dove il dominio della soluzione numerica è definito in $x \in (0, 2)$. In questo modo la lunghezza del passo spaziale viene calcolata come $\Delta x = \frac{|d|}{n-1}$, dove $|d|$ è la distanza tra l'estremo inferiore e l'estremo superiore del dominio della soluzione numerica, n il numero di punti della funzione d'onda discretizzata. Il numero totale di passi temporali i è invece costante. Poniamo inoltre delle condizioni al contorno su x in modo tale da ottenere il primo punto della funzione invariato per tutto il calcolo:

$$u(0, t) = c \quad (21)$$

dove c è una costante e il suo valore è pari al valore del primo punto della funzione.

Burgers

L'equazione di Burgers unidimensionale è un'equazione differenziale alle derivate parziali:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}. \quad (22)$$

L'equazione per fornire la soluzione numerica del problema è data da:

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x} (u_i^n - u_{i-1}^n) + \nu \frac{\Delta t}{\Delta x^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n) \quad (23)$$

dove:

- ν è il coefficiente di diffusione dell'onda;

- u :

$$-\frac{2\nu \left(\frac{(-8t+2x)e^{-\frac{(-4t+x)^2}{4\nu(t+1)}}}{4\nu(t+1)} - \frac{(-8t+2x-4\pi)e^{-\frac{(-4t+x-2\pi)^2}{4\nu(t+1)}}}{4\nu(t+1)} \right)}{e^{-\frac{(-4t+x-2\pi)^2}{4\nu(t+1)}} + e^{-\frac{(-4t+x)^2}{4\nu(t+1)}}} + 4 \quad (24)$$

Il passo spaziale Δx viene calcolato come precedentemente mostrato per l'equazione di convezione. Il passo temporale Δt viene invece calcolato come $\Delta t = \frac{\sigma \Delta x^2}{n-1}$, dove σ è la costante di Courant-Friedrichs-Lowy (CFL) e n il numero di punti della funzione d'onda discretizzata. Il numero di passi temporali totali è $\frac{T}{\Delta t}$, dove T è il tempo totale di simulazione.

Le condizioni iniziali sono definite:

$$u(x, 0). \quad (25)$$

Le condizioni al contorno sono:

$$u(0) = u(2\pi). \quad (26)$$

3 Progettazione dell'Algoritmo

3.1 Scelte di Progetto

3.1.1 Dati del problema

Gli insiemi finiti di numeri reali (codomini di ciascuna funzione) si prestano in modo naturale ad essere implementati mediante strutture dati lineari. Queste verranno allocate dinamicamente in quanto non si conosce a priori il numero di punti del dominio di ciascuna funzione, in maniera da non imporre limitazioni rispetto alla specifica del problema.

3.1.2 Stampa a schermo

Per guidare al meglio l'utente durante la fase di acquisizione dei dati, si è scelto di stampare a schermo una descrizione dei parametri utilizzati da ciascun modello di simulazione. Questi sono stati divisi in:

- parametri iniziali: sono dei valori costanti che vengono utilizzati per impostare le condizioni iniziali di ciascuna simulazione numerica;
- parametri di simulazione: sono dei valori costanti fissati a priori che limitano l'esperienza di personalizzazione della simulazione numerica dell'utente ai soli dati richiesti in ingresso;
- parametri richiesti all'utente: sono i dati che l'utente può inserire per ciascuna simulazione. Per ciascun dato viene fornita un'indicazione sui valori più appropriati da digitare.

3.2 Passi dell'Algoritmo

I passi dell'algoritmo per risolvere il problema sono i seguenti:

1. Acquisire la lunghezza del passo temporale per l'equazione di moto fugoide senza attrito: **v1**.
2. Calcolare e stampare l'integrazione numerica dell'equazione di moto fugoide senza attrito: **2.1**.
3. Acquisire la lunghezza del passo temporale per l'equazione di moto fugoide con attrito: **v1**.
4. Calcolare e stampare l'integrazione numerica dell'equazione di moto fugoide con attrito: **4.1**.
5. Acquisire il numero di punti totali della funzione d'onda per l'equazione di convezione lineare unidimensionale: **v2**. e **5.1**.
6. Calcolare e stampare l'integrazione numerica dell'equazione di convezione lineare unidimensionale: **6.1**.
7. Acquisire il numero di punti totali della funzione d'onda per l'equazione di Burgers unidimensionale: **v2**.
8. Calcolare e stampare l'integrazione numerica dell'equazione di Burgers unidimensionale: **8.1**.

2.1 Calcolo del moto fugoide senza attrito

- Alla condizione iniziale z_0 si aggiunge il resto del calcolo per l'integrazione numerica dell'equazione:
 - Caso base: se il numero di passi temporali è pari a zero, si effettua un passo di integrazione numerica (vedi metodo di Eulero, in 2.3, 6).
 - Caso generale: se il numero di passi temporali è maggiore di zero, si effettua un passo di integrazione numerica e poi, una volta decrementato di uno il numero di passi temporali, si procede ricorsivamente sul numero di passi rimanenti.

4.1 Calcolo del moto fugoide con attrito

- Alla condizione iniziale y_0 si aggiunge il resto del calcolo per l'integrazione numerica dell'equazione:
 - Caso base: se il numero di passi temporali è pari a zero, si effettua un passo di integrazione numerica (vedi metodo di Eulero, in 2.3, 17).
 - Caso generale: se il numero di passi temporali è maggiore di zero, si effettua un passo di integrazione numerica e poi, una volta decrementato di uno il numero di passi temporali, si procede ricorsivamente sul numero di passi rimanenti.

5.1 Acquisizione dati di convezione

- Se il numero di punti è pari a zero o uno, viene calcolata solamente la condizione iniziale: **6.1**
- Se il numero di punti è maggiore di uno, si acquisisce la lunghezza del passo temporale (**v1.**), si calcola la condizione iniziale (**6.1**) e si procede con l'integrazione numerica: **6.2**

6.1 Calcolo della condizione iniziale per l'equazione di convezione

- Si genera una lista di punti equidistanti fra loro rappresentante il dominio della funzione d'onda quadra:
 - a.
- Si calcola su ogni punto del dominio la funzione d'onda quadra (vedi onda u , in 2.3, 20).

6.2 Calcolo dell'integrazione numerica per l'equazione di convezione

- Si calcola numericamente l'integrazione della funzione rispetto al tempo:
 - Caso base: se il numero di passi temporali è uguale a zero, viene restituita la funzione d'onda quadra.
 - Caso generale: se il numero di passi temporali è maggiore di zero, si decrementa di uno il numero di passi temporali, si calcola la condizione al contorno (vedi 2.3, 21) e la si aggiunge in testa al calcolo numerico dell'integrazione della funzione rispetto allo spazio. Quest'ultima viene effettuata come segue:
 - * Caso base: se si raggiunge il numero di passi spaziali totale, si effettua un passo di integrazione numerica (vedi metodo di Eulero, in 2.3, 19).
 - * Caso generale: se il numero di passi spaziali complessivo non è stato ancora raggiunto, si effettua un passo di integrazione numerica e poi, una volta incrementato di uno il numero di passi spaziali effettuati, si procede ricorsivamente sul numero di passi rimanenti.

8.1 Calcolo dell'equazione di Burgers

- Se il numero di punti è zero o uno, l'integrazione numerica dell'equazione è uguale al calcolo della condizione iniziale: **8.2**.
- Se il numero di punti è maggiore di uno, si calcola la condizione iniziale dell'equazione (**8.2**) e si procede con la sua integrazione numerica: **8.3**.

8.2 Calcolo della condizione iniziale per l'equazione di Burgers

- Si genera una lista di punti equidistanti fra loro rappresentante il dominio della funzione d'onda a dente di sega: **a**.
- Si applica la condizione iniziale (vedi 2.3, 25) alla funzione d'onda a dente di sega (vedi onda u , in 2.3, 24).

8.3 Calcolo dell'integrazione numerica per l'equazione di Burgers

- Si calcola numericamente l'integrazione della funzione rispetto al tempo:
 - Caso base: se il numero di passi temporali è uguale a zero, viene restituita la funzione d'onda a dente di sega.
 - Caso generale: se il numero di passi temporali è maggiore di zero, si decrementa di uno il numero di passi temporali, si calcola la condizione al contorno (vedi 2.3, 26) e la si aggiunge in testa al calcolo numerico dell'integrazione della funzione rispetto allo spazio. Quest'ultima viene effettuata come segue:
 - * Caso base: se si raggiunge il numero di passi spaziali totale, si calcola la condizione di bordo (vedi 2.3, 26).
 - * Caso generale: se il numero di passi spaziali complessivo non è stato ancora raggiunto, si effettua un passo di integrazione numerica (vedi metodo di Eulero, in 2.3, 23) e poi, una volta incrementato di uno il numero di passi spaziali effettuati, si procede ricorsivamente sul numero di passi rimanenti.

a. Generazione di punti equidistanti

- Se il numero di punti da generare è zero, si restituisce una sequenza di punti è vuota.
- Se il numero di punti da generare è maggiore di zero, si calcola la distanza tra l'estremo superiore e quello inferiore del dominio, si decrementa di uno il numero di punti totali e partendo dall'estremo inferiore del dominio:
 - Caso base: se si è raggiunto il numero massimo di punti da generare, si aggiunge alla sequenza di punti l'estremo inferiore del dominio.
 - Caso generale: se il numero massimo di punti da generare non è stato ancora raggiunto, si aggiunge alla sequenza il punto dell'estremo inferiore del dominio, si incrementa di uno il numero di punti calcolati, si calcola il punto successivo che diventa il nuovo estremo inferiore del dominio e si procede ricorsivamente sul numero di punti da generare rimasti.

v1. Validazione dell'acquisizione della lunghezza del passo temporale

- Caso base: se il valore del passo temporale è maggiore di zero, allora il dato viene acquisito.
- Caso generale: altrimenti si stampa su schermo un messaggio di errore e viene ripetuta l'acquisizione.

v2. Validazione dell'acquisizione del numero di punti di una funzione

- Caso base: se il numero di punti è un intero positivo, il valore viene acquisito.
- Caso generale: altrimenti si stampa su schermo un messaggio di errore e viene ripetuta l'acquisizione.

4 Implementazione dell'algoritmo

4.1 Haskell

File sorgente

```
{- Programma Haskell per effettuare simulazioni numeriche -}
import Data.List {- necessario per usare:
                  - !!,      che estrae l'n-esimo elemento di una lista;
                  - head,    che estrae il primo elemento di una lista;
                  - last,    che estrae l'ultimo elemento di una lista;
                  - tail,    che estrae gli elementi di una lista successivi al primo;
                  - init,    che estrae gli elementi di una lista precedenti all'ultimo;
                  - abs,     che ritorna il valore assoluto di un numero;
                  - reverse, che ritorna una lista di elementi in ordine inverso;
                  - length,  che ritorna la lunghezza della lista. -}

{- Costanti globali -}

{- Limiti del dominio spaziale dell'equazione di convezione -}
inf_conv = 0.0 -- Estremo inferiore del dominio spaziale.
sup_conv = 2.0 -- Estremo superiore del dominio spaziale.

{- Forza gravitazionale terrestre. -}
c_grv :: Double
c_grv = 9.81

{- Coefficiente di diffusione -}
nu :: Double
nu = 0.07

{- Velocita' di trim del velivolo, misurato in m/s -}
v_trim :: Double
v_trim = 30.0

{- Tipo di dato che rappresenta una coppia di elementi uguali. -}
type Coppia a = (a,a)

{- Tipo di dato che rappresenta una quadrupla di elementi uguali. -}
type Quadrupla a = (a,a,a,a)

main :: IO()
main = do putStrLn "Progetto della sessione estiva del corso Programmazione Logica e Funzionale"
          putStrLn "Anno 2023/2024"
          putStrLn "Corso tenuto dal prof. Marco Bernardo"
          putStrLn "Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele"
          putStrLn "-----"
          putStrLn "| Calcolo del moto fugoide senza attrito"
          putStrLn "| Parametri iniziali:"
          putStrLn "| altitudine iniziale      = 100m,"
          putStrLn "| velocita' iniziale       = 10m/s."
          putStrLn "| Parametri di simulazione:
```

```

putStrLn "| secondi di simulazione = 100s,
putStrLn "| costante gravitazionale = 9.81m/(s^2).
putStrLn "| Parametro richiesto:
putStrLn "| passo temporale, determina la distanza temporale
putStrLn "| tra due punti di simulazione, un valore basso
putStrLn "| permette una simulazione piu' accurata.
putStrLn "| Il valore del passo temporale deve essere maggiore di zero.
putStrLn "-----"

```

dt <- acquisisci_dato_dt
putStrLn \$ show (calc_fugoide_semplice (read dt :: Double))

```

putStrLn "-----"
putStrLn "| Calcolo del moto fugoide con attrito
putStrLn "| Parametri iniziali:
putStrLn "| velocita' iniziale = velocita' di trim,
putStrLn "| angolo iniziale = Orad,
putStrLn "| spostamento laterale iniziale = 0m,
putStrLn "| spostamento verticale iniziale = 1000m.
putStrLn "| Parametri di simulazione:
putStrLn "| secondi di simulazione = 100s,
putStrLn "| velocita' di trim = 30m/s,
putStrLn "| costante gravitazionale = 9.81m/(s^2),
putStrLn "| coefficiente di resistenza dell'aria = 0.025,
putStrLn "| coefficiente di portanza = 1N.
putStrLn "| Parametro richiesto:
putStrLn "| passo temporale, determina la distanza temporale
putStrLn "| tra due punti di simulazione, un valore basso
putStrLn "| permette una simulazione piu' accurata.
putStrLn "| Il valore del passo temporale deve essere maggiore di zero.
putStrLn "-----"

```

dt <- acquisisci_dato_dt
putStrLn \$ show (calc_fugoide_completo (read dt :: Double))

```

putStrLn "-----"
putStrLn "| Calcolo dell'equazione di convezione lineare a una dimensione
putStrLn "| Parametri iniziali:
putStrLn "| estremo superiore del dominio spaziale = 2.0,
putStrLn "| estremo inferiore del dominio spaziale = 0.0,
putStrLn "| valore della parte alta della funzione = 2.0,
putStrLn "| valore della parte bassa della funzione = 1.0.
putStrLn "| Parametri di simulazione:
putStrLn "| numero di passi temporali da effettuare = 25,
putStrLn "| velocita' dell'onda = 1.0.
putStrLn "| Parametri richiesti all'utente:
putStrLn "| numero di punti che compongono la funzione d'onda,
putStrLn "| un valore alto permette una simulazione piu' accurata.
putStrLn "| Il valore del numero di punti dell'onda deve essere un naturale.
putStrLn "| Passo temporale, determina la distanza temporale
putStrLn "| tra due punti di simulazione, un valore basso
putStrLn "| permette una simulazione piu' accurata.
putStrLn "| Il valore del passo temporale deve essere maggiore di zero.
putStrLn "-----"

```

acqui_dati_e_calc_conv

```

putStrLn "-----"
putStrLn "| Calcolo dell'equazione di Burgers a una dimensione |"
putStrLn "| Parametri iniziali: |"
putStrLn "| estremo superiore del dominio spaziale = 2.0 * pi, |"
putStrLn "| estremo inferiore del dominio spaziale = 0.0. |"
putStrLn "| Parametri di simulazione: |"
putStrLn "| tempo finale di simulazione = 0.6s, |"
putStrLn "| coefficiente di diffusione = 1.0m^2/s, |"
putStrLn "| costante di Courant-Friedrichs-Lowy = 0.1. |"
putStrLn "| Parametri richiesti all'utente: |"
putStrLn "| numero di punti che compongono la funzione d'onda. |"
putStrLn "| Il valore del numero di punti dell'onda deve essere un naturale. |"
putStrLn "-----"

nx <- acquisisci_dato_nxb
putStrLn $ show (calc_burgers (read nx :: Int))

{- Inizio sezione input/output -}

{- L'azione input/output acquisisci_dato_dt acquisisce un parametro numerico
reale di simulazione, ovvero la lunghezza del passo temporale.-}

acquisisci_dato_dt :: IO String
acquisisci_dato_dt = do putStrLn "Digita lunghezza del passo temporale: "
                        dt <- getLine
                        if ((read dt :: Double) > 0)
                            then return dt
                        else do putStrLn "Acquisizione errata!"
                                putStrLn "Il valore deve essere maggiore di zero."
                                acquisiisci_dato_dt

{- L'azione input/output acqui_dati_e_calc_conv acquisisce due parametri numerici
di simulazione per l'equazione di convezione: il primo, un naturale per il
numero di punti della funzione d'onda; il secondo un reale per la lunghezza
del passo temporale. In seguito all'acquisizione dei parametri calcola l'integrazione
numerica dell'equazione di convezione. -}

acqui_dati_e_calc_conv :: IO ()
acqui_dati_e_calc_conv = do putStrLn "Digita il numero di punti totali della funzione d'onda: "
                           nx <- getLine
                           if (((read nx :: Integer) == 0) ||
                               ((read nx :: Integer) == 1))
                               then putStrLn $ show (cond_iniziale (read nx :: Int) onda_quadra inf_onda)
                           else if ((read nx :: Integer) > 1)
                               then do dt <- acquisisci_dato_dt
                                       putStrLn $ show (calc_convezione (read nx :: Int) (read dt :: Double))
                               else do putStrLn "Acquisizione errata!"
                                       putStrLn "Il valore deve essere un numero naturale."
                                       acqui_dati_e_calc_conv

{- L'azione input/output acquisisci_dato_nxb acquisisce un parametro numerico

```

```

naturale di simulazione, ovvero il numero totale di punti della funzione
d'onda per il calcolo dell'equazione di Burgers. -}

acquisisci_dato_nxb :: IO String
acquisisci_dato_nxb = do putStrLn "Digita il numero di punti totali della funzione d'onda: "
    nx <- getLine
    if ((read nx :: Integer) >= 0)
        then return nx
    else do putStrLn "Acquisizione errata!"
            putStrLn "Il valore deve essere un numero naturale."
            acquisisci_dato_nxb

{- Fine sezione input/output -}

{- Inizio sezione fugoide semplice -}

{- La funzione calc_fugoide_semplice calcola il moto fugoide privo di attrito di
un velivolo generico:
- il suo unico argomento e' la lunghezza del passo temporale dt. -}

calc_fugoide_semplice :: Double -> [Double]
calc_fugoide_semplice dt = z0 : calc_moto_semplice (z0, b0) dt passi_temporali
    where
        z0          = 100.0           -- Altitudine iniziale del velivolo.
        b0          = 10.0            -- Velocita' iniziale del velivolo.
        tempo       = 100.0           -- Numero di secondi di simulazione.
        passi_temporali = floor(tempo/dt) + 1 -- Numero di punti in cui effettuare il calcolo.

{- La funzione calc_moto_semplice calcola numericamente l'integrazione del moto fugoide:
- il primo argomento e' una coppia di valori, ovvero altitudine e velocita'
del velivolo;
- il secondo argomento e' la lunghezza del passo temporale dt;
- il terzo argomento e' il numero di passi che sono ancora da effettuare. -}

calc_moto_semplice :: (Double, Double) -> Double -> Int -> [Double]
calc_moto_semplice (dAA, _) dt len | len == 0  = [dAA]
                                     | otherwise = dBA : calc_moto_semplice dB dt (len - 1)
    where
        dB@(dB, _) = passo_eulero_semplice dA dt

{- La funzione passo_eulero_semplice applica il metodo di Eulero ad una coppia di numeri. La
funzione approssima la soluzione al tempo t_(n+1) tramite il valore della funzione
al tempo t_n ed un opportuno passo temporale:
- il primo argomento e' una coppia di valori, ovvero la posizione e direzione del
velivolo al momento t_n;
- il secondo argomento e' la lunghezza del passo temporale dt. -}

passo_eulero_semplice :: (Double, Double) -> Double -> (Double, Double)
passo_eulero_semplice (y@alt, v@vel) dt = somma_coppia dA (molt_coppia_scalare (derivata_u_semplice

```

```

{- La funzione derivata_u_semplice viene utilizzata per l'applicazione dell'equazione del moto fugoide
 - il suo unico argomento e' una coppia di valori, ovvero altitudine e velocita'
   del velivolo. -}

derivata_u_semplice :: Coppia Double -> Coppia Double
derivata_u_semplice dA@(y@alt, v@vel) = (v, c_grv * (1-y/zt))
where
  zt = 100.0 -- Altitudine centrale all'oscillazione.

{- Fine sezione fugoide semplice -}

{- Inizio sezione fugoide completo -}

{- La funzione calc_fugoide_completo calcola il moto fugoide con attrito di
un velivolo generico:
 - il suo unico argomento e' la lunghezza del passo temporale dt. -}

calc_fugoide_completo :: Double -> [[Double]]
calc_fugoide_completo dt = [x0, y0] : calc_moto_completo (v0, theta0, x0, y0) dt passi_temporali
where
  v0          = v_trim           -- La velocita' iniziale, in questo caso quella di trim.
  theta0      = 0.0              -- Angolo iniziale del velivolo.
  x0          = 0.0              -- Spostamento orizzontale iniziale del velivolo.
  y0          = 1000.0            -- Altitudine iniziale del velivolo.
  tempo       = 100.0             -- Numero di secondi di simulazione.
  passi_temporali = floor(tempo/dt) + 1 -- Numero di punti in cui effettuare il calcolo.

{- La funzione calc_moto_completo calcola numericamente l'integrazione del moto fugoide:
 - il primo argomento e' una quadrupla di valori, ovvero velocita', angolo,
   spostamento laterale e verticale del velivolo;
 - il secondo argomento e' la lunghezza del passo temporale dt;
 - il terzo argomento e' il numero di passi che sono ancora da effettuare. -}

calc_moto_completo :: Quadrupla Double -> Double -> Int -> [[Double]]
calc_moto_completo dA dt len | len == 0  = [[dBC, dBd]]
                             | otherwise = [dBC, dBd] : calc_moto_completo dB dt (len - 1)
where
  dB@(_, _, dBC, dBd) = passo_eulero_completo dA dt

{- La funzione passo_eulero_completo applica il metodo di Eulero ad una quadrupla di numeri. La
funzione approssima la soluzione al tempo t_(n+1) tramite il valore della funzione
al tempo t_n ed un opportuno passo temporale:
 - il primo argomento e' una quadrupla di valori, ovvero la velocita', angolo, spostamento
   laterale e verticale del velivolo al momento t_n;
 - il secondo argomento e' la lunghezza del passo temporale dt. -}

passo_eulero_completo :: Quadrupla Double -> Double -> Quadrupla Double
passo_eulero_completo dA@(v, theta, x, y) dt = somma_quadrupla dA (molt_quadrupla_scalare (derivata_u_completo dA) dt)

{- La funzione derivata_u_completo viene utilizzata per l'applicazione dell'equazione del moto fugoide

```

```

- il suo unico argomento e' una coppia di valori, ovvero la velocita' e l'angolo del velivolo. -}

derivata_u_completo :: Coppia Double -> Quadrupla Double
derivata_u_completo dA@(v,theta) = (- (c_grv * sin theta) - (c_res / c_prt)*c_grv/v Trim**2*v**2,
                                     - (c_grv * cos theta / v) + c_grv/v Trim**2*v,
                                     v*cos theta,
                                     v*sin theta)

where
  c_res = 0.025 -- Coefficiente di resistenza all'aria.
  c_prt = 1.0   -- Coefficiente di portanza.

{- Fine sezione fugoide completo -}

{- Inizio sezione funzioni ausiliarie -}

{- La funzione somma_coppia prende due coppie dello stesso tipo e
  restituisce la coppia risultante dalla somma rispettiva degli elementi. -}

somma_coppia :: (Num a) => Coppia a -> Coppia a -> Coppia a
somma_coppia (a1,b1) (a2,b2) = (a1+a2, b1+b2)

{- La funzione molt_coppia_scalare prende una coppia di elementi numerici
  ed un valore numerico e restituisce la coppia risultante dalla
  moltiplicazione rispettiva degli elementi:
  - il primo argomento e' una coppia di elementi;
  - il secondo argomento e' un valore numerico. -}

molt_coppia_scalare :: (Num a) => Coppia a -> a -> Coppia a
molt_coppia_scalare (a1,b1) b = (a1*b, b1*b)

{- La funzione somma_quadrupla prende due quadruple dello stesso tipo e
  restituisce la quadrupla risultante dalla somma rispettiva degli elementi. -}

somma_quadrupla :: (Num a) => Quadrupla a -> Quadrupla a -> Quadrupla a
somma_quadrupla (a1,b1,c1,d1) (a2,b2,c2,d2) = (a1+a2, b1+b2, c1+c2, d1+d2)

{- La funzione molt_quadrupla_scalare prende una quadrupla di elementi numerici
  ed un valore numerico e restituisce la quadrupla risultante dalla
  moltiplicazione rispettiva degli elementi:
  - il primo argomento e' una quadrupla di elementi;
  - il secondo argomento e' un valore numerico. -}

molt_quadrupla_scalare :: (Num a) => Quadrupla a -> a -> Quadrupla a
molt_quadrupla_scalare (a1,b1,c1,d1) b = (a1*b, b1*b, c1*b, d1*b)

{- Fine sezione funzioni ausiliarie -}

{- Inizio sezione convezione lineare -}

```

```

{- La funzione calc_convezione calcola l'integrazione numerica
dell'equazione di convezione lineare a una dimensione:
- il primo argomento e' il numero di punti totali della funzione
d'onda;
- il secondo argomento e' la lunghezza del passo temporale. -}

calc_convezione :: Int -> Double -> [Double]
calc_convezione nx dt = tempo_conv nt nx c dx dt (cond_iniziale nx onda_quadra inf_conv sup_conv)
where
  nt = 25
  dx = abs(sup_conv - inf_conv) / (fromIntegral(nx :: Int) - 1)
  c = 1.0
  -- Numero complessivo di passi temporali
  -- Distanza tra qualsiasi coppia di punti
  -- Velocita' dell'onda.

{- La funzione tempo_conv calcola numericamente l'integrazione della
funzione rispetto al parametro temporale dt:
- il primo argomento e' il numero di passi temporali totali che la
funzione d'onda deve compiere;
- il secondo argomento e' il numero di passi spaziali utilizzati dal
predicato spazio_conv;
- il terzo argomento e' la costante di velocità dell'onda;
- il quarto argomento e' la lunghezza del passo spaziale;
- il quinto argomento e' la lunghezza del passo temporale;
- il sesto argomento e' la funzione d'onda ricalcolata. -}

tempo_conv :: Int -> Int -> Double -> Double -> Double -> [Double] -> [Double]
tempo_conv 0 _ _ _ _ onda      = onda
tempo_conv nt nx c dx dt onda = tempo_conv (nt - 1) nx c dx dt ((head onda) : spazio_conv 1 c dx dt onda)

{- La funzione spazio_conv calcola numericamente l'integrazione della
funzione rispetto al parametro spaziale dx:
- il primo argomento e' il numero di passi temporali che la funzione
d'onda ha compiuto;
- il secondo argomento e' la costante di velocità dell'onda;
- il terzo argomento e' la lunghezza del passo spaziale;
- il quarto argomento e' la lunghezza del passo temporale;
- il quinto argomento e' la funzione d'onda. -}

spazio_conv :: Int -> Double -> Double -> Double -> Double -> [Double] -> [Double]
spazio_conv i c dx dt lx | i == length lx - 1 = [passo_eulero]
                         | otherwise           = passo_eulero : spazio_conv (i+1) c dx dt lx
where
  passo_eulero = lx !! i - c * dt / dx * (lx !! i - lx !! (i-1))

{- La funzione onda_quadra calcola la funzione d'onda quadra:
- il suo unico argomento e' la lista di punti equidistanti del dominio
spaziale. -}

onda_quadra :: Double -> Double
onda_quadra x | x >= 0.5 && x <= 1.0 = onda_sup
               | otherwise           = onda_inf
where

```

```

onda_sup = 2.0 -- Valori assunti dalla parte alta della funzione d'onda quadra.
onda_inf = 1.0 -- Valori assunti dalla parte bassa della funzione d'onda quadra.

{- Fine sezione convezione lineare -}

{- Inizio sezione equazione di Burgers -}

{- La funzione calc_burgers calcola l'integrazione numerica
dell'equazione di Burgers a una dimensione:
- il suo unico argomento e' il numero di punti totali della funzione
d'onda. -}

calc_burgers :: Int -> [Double]
calc_burgers nx | nx == 0 || nx == 1 = cond_iniziale nx onda_dente_sega inf sup
                | otherwise           = tempo_burg nt nx dx dt (cond_iniziale nx onda_dente_sega inf sup)
where
    inf      = 0.0                                -- Estremo inferiore del dominio spaziale.
    sup      = 2.0 * pi                            -- Estremo superiore del dominio spaziale.
    dx       = abs(sup - inf) / (fromIntegral(nx :: Int) - 1) -- Distanza tra qualsiasi coppia di punti.
    sigma    = 0.1                                -- Costante di Courant-Friedrichs-Lowy.
    dt       = sigma * dx**2 / nu                  -- Lunghezza del passo temporale.
    t_fine   = 0.6                                -- Tempo totale di simulazione.
    nt       = floor(t_fine / dt)                  -- Numero complessivo di passi temporali.

{- La funzione tempo_burg calcola numericamente l'integrazione della
funzione rispetto al parametro temporale dt:
- il primo argomento e' il numero di passi temporali totali che la
funzione d'onda deve compiere;
- il secondo argomento e' il numero di passi spaziali utilizzati dalla
funzione spazio_burg;
- il terzo argomento e' la lunghezza del passo spaziale;
- il quarto argomento e' la lunghezza del passo temporale;
- il quinto argomento e' la funzione d'onda ricalcolata. -}

tempo_burg :: Int -> Int -> Double -> Double -> [Double] -> [Double]
tempo_burg 0 _ _ _ onda      = onda
tempo_burg nt nx dx dt onda = tempo_burg (nt - 1) nx dx dt (bordo : spazio_burg 1 dx dt onda)
where
    bordo = head onda - head onda * dt/dx * (head onda - last onda) +
             nu*dt/dx**2 * ((head $ tail onda) - 2*(head onda) + last onda) -- Condizione di bordo.

{- La funzione spazio_burg calcola numericamente l'integrazione della
funzione rispetto al parametro spaziale dx:
- il primo argomento e' l'indice per accedere agli elementi della
lista, viene utilizzato da passo_eulero;
- il secondo argomento e' la lunghezza del passo spaziale;
- il terzo argomento e' la lunghezza del passo temporale;
- il quarto argomento e' la funzione d'onda. -}

spazio_burg :: Int -> Double -> Double -> [Double] -> [Double]
spazio_burg i dx dt lx | i == length lx - 1 = [bordo]

```

```

| otherwise          = (passo_eulero) : spazio_burg (i+1) dx dt lx
where
bordo      = (last lx) - (last lx) * dt/dx * ((last lx) - (last $ init lx)) +
nu*dt/dx**2*(head lx - 2*(last lx) + (last $ init lx))
passo_eulero = (lx !! i - lx !! i * dt/dx * (lx !! i - lx !! (i-1)) +
nu*dt/dx**2*(lx !! (i+1) - 2*(lx !! i) + lx !! (i-1)))

{- La funzione onda_dente_sega calcola la funzione d'onda a dente di
sega 'u':
- il suo unico argomento e' la lista di punti equidistanti del
dominio spaziale.
Per il calcolo dell'onda si fa uso dei predicati phi e phi_primo
che sono rispettivamente la funzione e la sua derivata. -}

onda_dente_sega :: Double -> Double
onda_dente_sega x = u t0 x nu
where
t0 = 0.0
phi_primo t0 x nu = (-(-8*t0 + 2*x)*exp(-(-4*t0 + x)**2/(4*nu*(t0 + 1)))/(4*nu*(t0 + 1)) -
(-8*t0 + 2*x - 4*pi)*exp(-(-4*t0 + x - 2*pi)**2/(4*nu*(t0 + 1)))/(4*nu*(t0 + 1)))
phi t0 x nu      = exp(-(x-4*t0)**2/(4*nu*(t0+1))) + exp(-(x-4*t0-2*pi)**2/(4*nu*(t0+1)))
u t0 x nu        = -2*nu*((phi_primo t0 x nu) / (phi t0 x nu))+4

{- Fine sezione equazione di Burgers -}

{- Inizio sezione funzioni condivise -}

{- La funzione cond_iniziale calcola la condizione iniziale (una funzione)
per l'integrazione numerica dell'equazione di convezione e di Burgers:
- il primo argomento e' il numero di punti della griglia spaziale;
- il secondo argomento e' la funzione onda_quadra (onda_dente_sega)
utilizzata per il calcolo delle omonime funzioni;
- il terzo argomento e' l'estremo inferiore del dominio spaziale;
- il quarto argomento e' l'estremo superiore del dominio spaziale. -}

cond_iniziale :: Int -> (Double -> Double) -> Double -> Double -> [Double]
cond_iniziale nx onda inf sup = [onda x | x <- lx]
where
lx = gen_punti_equi nx inf sup

{- La funzione gen_punti_equi genera una lista di punti equidistanti tra loro:
- il primo argomento e' il numero di punti che si vuole generare;
- il secondo argomento e' l'estremo inferiore della lista di punti;
- il terzo argomento e' l'estremo superiore della lista di punti.
Per il calcolo dei punti si fa uso della funzione calc_punti. -}

gen_punti_equi :: Int -> Double -> Double -> [Double]
gen_punti_equi 0 _ _           = []
gen_punti_equi nx inf sup | sup > inf = calc_punti 0 (nx - 1) inf (abs $ sup - inf)
                           | otherwise = reverse(calc_punti 0 (nx - 1) sup (abs $ sup - inf))

```

```
calc_punti :: Int -> Int -> Double -> Double -> [Double]
calc_punti i nx inf dst | i == nx    = [inf]
                         | otherwise = inf : calc_punti (i + 1) nx (inf + (dst / fromIntegral(nx :: Int))

{- Fine sezione funzioni condivise -}
```

4.2 Prolog

File sorgente

```
/* Programma Prolog per effettuare simulazioni numeriche */

main :-
    write('Progetto della sessione estiva del corso Programmazione Logica e Funzionale'), nl,
    write('Anno 2023/2024'), nl,
    write('Corso tenuto dal prof. Marco Bernardo'), nl,
    write('Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele'), nl,

    write('-----'), nl,
    write('| Calcolo del moto fugoide senza attrito      |'), nl,
    write('| Parametri iniziali:                         |'), nl,
    write('| altitudine iniziale = 100m,                  |'), nl,
    write('| velocita\' iniziale = 10m/s.                 |'), nl,
    write('| Parametri di simulazione:                   |'), nl,
    write('| secondi di simulazione = 100s,                |'), nl,
    write('| costante gravitazionale = 9.81m/(s^2).     |'), nl,
    write('| Parametro richiesto:                        |'), nl,
    write('| passo temporale, determina la distanza temporale |'), nl,
    write('| tra due punti di simulazione, un valore basso |'), nl,
    write('| permette una simulazione piu\' accurata.   |'), nl,
    write('| Il valore del passo temporale deve essere maggiore di zero. |'), nl,
    write('-----'), nl,

    acquisisisci_dato_dt(DT),
    calc_fugoide_semplice(DT, FS),
    write(FS), nl,

    write('-----'), nl,
    write('| Calcolo del moto fugoide con attrito      |'), nl,
    write('| Parametri iniziali:                         |'), nl,
    write('| velocita\' iniziale = velocita\' di trim,    |'), nl,
    write('| angolo iniziale = 0rad,                      |'), nl,
    write('| spostamento laterale iniziale = 0m,          |'), nl,
    write('| spostamento verticale iniziale = 1000m.     |'), nl,
    write('| Parametri di simulazione:                   |'), nl,
    write('| secondi di simulazione = 100s,                |'), nl,
    write('| velocita\' di trim = 30m/s,                  |'), nl,
    write('| costante gravitazionale = 9.81m/(s^2),     |'), nl,
    write('| coefficiente di resistenza dell\'aria = 0.025, |'), nl,
    write('| coefficiente di portanza = 1N.               |'), nl,
    write('| Parametro richiesto:                        |'), nl,
    write('| passo temporale, determina la distanza temporale |'), nl,
    write('| tra due punti di simulazione, un valore basso |'), nl,
    write('| permette una simulazione piu\' accurata.   |'), nl,
    write('| Il valore del passo temporale deve essere maggiore di zero. |'), nl,
    write('-----'), nl,

    acquisisisci_dato_dt(DT1),
    calc_fugoide_completo(DT1, FC),
    write(FC), nl,

    write('-----'), nl,
```

```

write('' Calcolo dell\' equazione di convezione lineare a una dimensione |'), nl,
write('' Parametri iniziali: |'), nl,
write('' estremo superiore del dominio spaziale = 2.0, |'), nl,
write('' estremo inferiore del dominio spaziale = 0.0, |'), nl,
write('' valore della parte alta della funzione = 2.0, |'), nl,
write('' valore della parte bassa della funzione = 1.0. |'), nl,
write('' Parametri di simulazione: |'), nl,
write('' numero di passi temporali da effettuare = 25, |'), nl,
write('' velocita\ dell\'onda = 1.0. |'), nl,
write('' Parametri richiesti all\'utente: |'), nl,
write('' numero di punti che compongono la funzione d\'onda, |'), nl,
write('' un valore alto permette una simulazione piu\' accurata. |'), nl,
write('' Il valore del numero di punti dell\'onda deve essere un naturale. |'), nl,
write('' Passo temporale, determina la distanza temporale |'), nl,
write('' tra due punti di simulazione, un valore basso |'), nl,
write('' permette una simulazione piu\' accurata. |'), nl,
write('' Il valore del passo temporale deve essere maggiore di zero. |'), nl,
write('-----'), nl,

acquisisci_dati_conv(NXC,DT2),
calc_convezione(NXC,DT2,CONV),
write(CONV), nl,

write('-----'), nl,
write('' Calcolo dell\'equazione di Burgers a una dimensione |'), nl,
write('' Parametri iniziali: |'), nl,
write('' estremo superiore del dominio spaziale = 2.0 * pi, |'), nl,
write('' estremo inferiore del dominio spaziale = 0.0. |'), nl,
write('' Parametri di simulazione: |'), nl,
write('' tempo finale di simulazione = 0.6s, |'), nl,
write('' coefficiente di diffusione = 1.0m^2/s, |'), nl,
write('' costante di Courant-Friedrichs-Lowy = 0.1. |'), nl,
write('' Parametri richiesti all\'utente: |'), nl,
write('' numero di punti che compongono la funzione d\'onda. |'), nl,
write('' Il valore del numero di punti dell\'onda deve essere un naturale. |'), nl,
write('-----'), nl,

acquisisci_dato_nxb(NXB),
calc_burgers(NXB,BURG),
write(BURG).

/* Inizio sezione input/output */

/* Il predicato acquisisci_dato_dt acquisisce un parametro numerico
   reale positivo di simulazione, ovvero la lunghezza del passo temporale. */

acquisisci_dato_dt(DT) :- write('Digita lunghezza del passo temporale: '),
                      read(DTV),
                      DTV > 0,
                      DT is DTV,
                      !;
                      write('Acquisizione errata!'), nl,
                      write('Il valore deve essere maggiore di zero.'), nl,
                      acquisisci_dato_dt(DT).

```

```

/* Il predicato acquisisci_dati_conv acquisisce due parametri numerici
di simulazione per l'equazione di convezione: il primo, un naturale per il
numero di punti della funzione d'onda; il secondo un reale positivo
per la lunghezza del passo temporale. */

acquisisci_dati_conv(NX,DT) :- write('Digita il numero di punti totali della funzione d\'onda: '),
                                read(NXV),
                                integer(NXV),
                                acquisisci_dato_nxc(NXV,NX,DT),
                                !;
                                write('Acquisizione errata!'), nl,
                                write('Il valore deve essere un numero naturale.'), nl,
                                acquisisci_dati_conv(NX,DT).

acquisisci_dato_nxc(NX,NX,_) :- (NX == 0;
                                    NX == 1).
acquisisci_dato_nxc(NX,NX,DT) :- NX > 1,
                                acquisisci_dato_dt(DT).

/* Il predicato acquisisci_dato_nxb acquisisce un parametro numerico
naturale di simulazione, ovvero il numero totale di punti della funzione
d'onda per il calcolo dell'equazione di Burgers. */

acquisisci_dato_nxb(NX) :- write('Digita il numero di punti totali della funzione d\'onda: '),
                           read(NXV),
                           integer(NXV),
                           NXV >= 0,
                           NX is NXV,
                           !;
                           write('Acquisizione errata!'), nl,
                           write('Il valore deve essere un numero naturale.'), nl,
                           acquisisci_dato_nxb(NX).

/* Fine sezione input/output */

/* Inizio sezione fugoide semplice */

/* Il predicato calc_fugoide_semplice calcola il moto fugoide con attrito di
un velivolo generico:
- il primo argomento e' la lunghezza del passo temporale dt;
- il secondo argomento e' la funzione di traiettoria risultante. */

calc_fugoide_semplice(DT,[Z0|T]) :- Z0 is 100.0,                               /* Altitudine iniziale del velivolo.
                                         B0 is 10.0,                                /* Angolo iniziale del velivolo.
                                         PASSI is (floor(100.0/DT) + 1), /* Numero di punti in cui effettua
                                         calc_moto(Z0,B0,DT,PASSI,T).

/* Il predicato calc_moto calcola numericamente l'integrazione del moto fugoide:
- il primo argomento e' l'altitudine del velivolo;

```

```

- il secondo argomento e' la velocita' del velivolo;
- il terzo argomento e' la lunghezza del passo temporale dt;
- il quarto argomento e' il numero di passi che sono ancora da effettuare;
- il quinto argomento e' una lista di valori numerici che rappresentano
  l'altitudine del velivolo per ogni passo temporale. */

calc_moto(Y,V,DT,0,[YT])      :- passo_eulero(Y,V,DT,YT,_).
calc_moto(Y,V,DT,LEN,[YT|T])  :- LEN > 0,
                                passo_eulero(Y,V,DT,YT,VT),
                                LEN1 is (LEN - 1),
                                calc_moto(YT,VT,DT,LEN1,T).

/* Il predicato passo_eulero applica il metodo di Eulero ad una coppia di numeri. Il
predicato approssima la soluzione al tempo t_(n+1) tramite il valore del predicato
al tempo t_n ed un opportuno passo temporale:
- il primo argomento e' l'altitudine del velivolo;
- il secondo argomento e' la velocita' del velivolo;
- il terzo argomento e' la lunghezza del passo temporale dt;
- il quarto argomento e' l'altitudine del velivolo ricalcolata;
- il quinto argomento e' la velocita' del velivolo ricalcolata. */

passo_eulero(Y,V,DT,Y1,V1)  :- derivata_u(Y,V,YT,VT),
                                Y1 is (Y + (YT * DT)),
                                V1 is (V + (VT * DT)).

/* Il predicato derivata_u viene utilizzato per l'applicazione dell'equazione del
moto fugoide:
- il primo argomento e' l'altitudine del velivolo;
- il secondo argomento e' l'angolazione del velivolo;
- il terzo argomento e' l'altitudine del velivolo ricalcolata;
- il quarto argomento e' l'angolazione del velivolo ricalcolata. */

derivata_u(Y,V,V1)  :- CG is 9.81,      /* Costante gravitazionale terrestre. */
                      ZT is 100.0,    /* Altitudine centrale all'oscillazione. */
                      V1 is CG * (1-Y/ZT).

/* Fine sezione fugoide semplice */

/* Inizio sezione fugoide completo */

/* Il predicato calc_fugoide_completo calcola il moto fugoide con attrito di
un velivolo generico:
- il primo argomento e' la lunghezza del passo temporale dt;
- il secondo argomento e' la funzione di traiettoria risultante. */

calc_fugoide_completo(DT,[[X0|Y0]|T])  :- V0 is 30.0,                               /* La velocita' iniziale,
                                                THETA0 is 0.0,                         /* Angolo iniziale del vel
                                                X0 is 0.0,                             /* Spostamento orizzontale
                                                Y0 is 1000.0,                          /* Altitudine iniziale del
                                                PASSI is (floor(100.0/DT) + 1), /* Numero di punti in cui
                                                calc_moto(V0,THETA0,X0,Y0,DT,PASSI,T).

```

```

/* Il predicato calc_moto calcola numericamente l'integrazione del moto fugoide:
   - il primo argomento e' la velocita' del velivolo;
   - il secondo argomento e' l'angolo del velivolo;
   - il terzo argomento e' lo spostamento laterale del velivolo;
   - il quarto argomento e' lo spostamento verticale del velivolo;
   - il quinto argomento e' la lunghezza del passo temporale dt;
   - il sesto argomento e' il numero di passi che sono ancora da effettuare;
   - il settimo argomento e' una lista di coppie di valori numerici che rappresentano
     spostamento laterale e l'altitudine del velivolo per ogni passo temporale. */

calc_moto(V,THETA,X,Y,DT,0,[X1,Y1])      :- passo_eulero(V,THETA,X,Y,DT,_,_,X1,Y1).
calc_moto(V,THETA,X,Y,DT,LEN,[[X1|Y1]|T]) :- LEN > 0,
                                              passo_eulero(V,THETA,X,Y,DT,V1,THETA1,X1,Y1),
                                              LEN1 is (LEN - 1),
                                              calc_moto(V1,THETA1,X1,Y1,DT,LEN1,T).

/* Il predicato passo_eulero applica il metodo di Eulero ad una quadrupla di numeri. Il
predicato approssima la soluzione al tempo t_(n+1) tramite il valore del predicato
al tempo t_n ed un opportuno passo temporale:
   - il primo argomento e' la velocita' del velivolo;
   - il secondo argomento e' l'angolo del velivolo;
   - il terzo argomento e' lo spostamento laterale del velivolo;
   - il quarto argomento e' lo spostamento verticale del velivolo;
   - il quinto argomento e' la lunghezza del passo temporale dt;
   - il sesto argomento e' la velocita' del velivolo ricalcolata;
   - il settimo argomento e' l'angolo del velivolo ricalcolato;
   - il ottavo argomento e' lo spostamento laterale del velivolo ricalcolato;
   - il nono argomento e' lo spostamento verticale del velivolo ricalcolato. */

passo_eulero(V,THETA,X,Y,DT,V1,THETA1,X1,Y1) :- derivata_u(V,THETA,VT,THETAT,XT,YT),
                                              V1      is (V + (VT * DT)),
                                              THETA1  is (THETA + (THETAT * DT)),
                                              X1      is (X + (XT * DT)),
                                              Y1      is (Y + (YT * DT)).

/* Il predicato derivata_u viene utilizzato per l'applicazione dell'equazione del
moto fugoide:
   - il primo argomento e' la velocita' del velivolo;
   - il secondo argomento e' l'angolazione del velivolo;
   - il terzo argomento e' la velocita' del velivolo ricalcolata;
   - il quarto argomento e' l'angolazione del velivolo ricalcolata;
   - il quinto argomento e' lo spostamento laterale del velivolo ricalcolato;
   - il sesto argomento e' lo spostamento verticale del velivolo ricalcolato. */

derivata_u(V,THETA,V1,THETA1,X1,Y1) :- CG is 9.81,      /* Costante gravitazionale terrestre. */
                                              CR is 0.025,    /* Coefficiente di resistenza all'aria. */
                                              CP is 1.0,      /* Coefficiente di portanza. */
                                              VTRIM is 30.0,  /* Velocita' di trim del velivolo. */
                                              V1 is (- (CG * sin(THETA)) - (CR / CP) * CG/VTRIM**2*V**2),
                                              THETA1 is (- (CG * cos(THETA) / V) + (CG/VTRIM**2*V)),
                                              X1 is (V * cos(THETA)),
                                              Y1 is (V * sin(THETA)).

```

```

/* Fine sezione fugoide completo */

/* Inizio sezione convezione lineare */

/* Il predicato calc_convezione calcola l'integrazione numerica
dell'equazione di convezione lineare a una dimensione:
- il primo argomento e' il numero di punti totali della
funzione d'onda;
- il secondo argomento e' la lunghezza del passo temporale;
- il terzo argomento e' l'integrazione numerica completa dell'equa-
zione lineare unidimensionale di convezione. */

calc_convezione(NX,_,F) :- (NX == 0;
                            NX == 1),
                           INF is 0.0,          /* Limite inferiore del dominio spaziale. */
                           SUP is 2.0,          /* Limite superiore del dominio spaziale. */
                           cond_iniziale_conv(NX,INF,SUP,F).

calc_convezione(NX,DT,F) :- NX > 1,
                           NT is 25,           /* Numero complessivo di passi temporali
che deve effettuare l'algoritmo. */
                           NX1 is NX - 1,
                           C is 1.0,            /* Velocita' dell'onda. */
                           INF is 0.0,          /* Limite inferiore del dominio spaziale. */
                           SUP is 2.0,          /* Limite superiore del dominio spaziale. */
                           DX is abs(SUP - INF) / NX1, /* Distanza tra qualsiasi coppia di
punti della griglia adiacenti. */
                           cond_iniziale_conv(NX,INF,SUP,ONDA),
                           tempo_conv(0,NT,NX1,C,DX,DT,ONDA,F).

/* Il predicato tempo_conv calcola numericamente l'integrazione della
funzione rispetto al parametro temporale DT:
- il primo argomento e' il numero di passi temporali che la funzione
d'onda ha compiuto;
- il secondo argomento e' il numero di passi temporali totali che la
funzione d'onda deve compiere;
- il terzo argomento e' il numero di passi spaziali utilizzati dal
predicato spazio_conv;
- il quarto argomento e' la costante di velocità dell'onda;
- il quinto argomento e' la lunghezza del passo spaziale;
- il sesto argomento e' la lunghezza del passo temporale;
- il settimo argomento e' la funzione d'onda ricalcolata;
- l' ottavo argomento e' la funzione d'onda risultante. */

tempo_conv(NT,NT,_,_,_,_,F,F).
tempo_conv(I,NT,NX1,C,DX,DT,ONDA,F) :- I < NT,
                                         I1 is I + 1,
                                         testa(ONDA,T),
                                         spazio_conv(0,NX1,C,DX,DT,ONDA,F1),
                                         inserisci_elem(T,F1,R),
                                         tempo_conv(I1,NT,NX1,C,DX,DT,R,F).

```

```

/* Il predicato spazio_conv calcola numericamente l'integrazione della
funzione rispetto al parametro spaziale DX:
- il primo argomento e' il numero di passi temporali che la funzione
d'onda ha compiuto;
- il secondo argomento e' il numero di passi temporali totali che la
funzione d'onda deve compiere;
- il terzo argomento e' la costante di velocità dell'onda;
- il quarto argomento e' la lunghezza del passo spaziale;
- il quinto argomento e' la lunghezza del passo temporale;
- il sesto argomento e' la funzione d'onda;
- il settimo argomento e' la funzione d'onda ricalcolata con
passo_eulero_conv. */

spazio_conv(NX1,NX1,_,_,_,[],[]).
spazio_conv(I,NX1,C,DX,DT,[E0|LX],[E|T]) :- I < NX1,
                                              I1 is I + 1,
                                              testa(LX,E1),
                                              passo_eulero_conv(E0,E1,C,DX,DT,EU),
                                              E is EU,
                                              spazio_conv(I1,NX1,C,DX,DT,LX,T).

/* Il predicato passo_eulero_conv effettua il passo di Eulero:
- il primo argomento e' l'elemento corrente della lista;
- il secondo argomento e' l'elemento successivo a quello corrente della lista;
- il terzo argomento e' la costante di velocità dell'onda;
- il quarto argomento e' la lunghezza del passo spaziale;
- il quinto argomento e' la lunghezza del passo temporale;
- il sesto argomento e' il valore risultante del punto applicato il passo di Eulero. */

passo_eulero_conv(E0,E1,C,DX,DT,EU) :- EU is E1 - C * (DT/DX) * (E1 - E0).

/* Il predicato cond_iniziale_conv calcola la condizione iniziale (una funzione)
per l'integrazione numerica dell'equazione di convezione:
- il primo argomento e' il numero di punti della griglia spaziale;
- il secondo argomento e' l'estremo inferiore del dominio spaziale;
- il terzo argomento e' l'estremo superiore del dominio spaziale;
- il quarto argomento e' la funzione d'onda quadra. */

cond_iniziale_conv(NX,INF,SUP,ONDA) :- gen_punti_equi(NX,INF,SUP,L),
                                         onda_quadra(L,ONDA).

/* Il predicato onda_quadra calcola la funzione d'onda quadra:
- il primo argomento e' la lista di punti equidistanti del dominio
spaziale;
- il secondo argomento e' la funzione d'onda calcolata. */

onda_quadra([],[]).
onda_quadra([X|L1],[OSI|T]) :- X >= 0.5,
                               X =< 1.0,
                               OSI is 2.0, /* Valori assunti dalla parte alta della
                                             funzione d'onda quadra. */
                               onda_quadra(L1,T).

```

```

onda_quadra([X|L1],[OSI|T]) :- (X < 0.5;
                                  X > 1.0),
                                  OSI is 1.0, /* Valori assunti dalla parte bassa della
                                                funzione d'onda quadra. */
                                  onda_quadra(L1,T).

/* Fine sezione convezione lineare */

/* Inizio sezione equazione di Burgers */

/* Il predicato calc_burgers calcola l'integrazione numerica
dell'equazione di Burgers a una dimensione:
- il primo argomento e' il numero di punti totali della funzione
d'onda;
- il secondo argomento e' l'integrazione numerica completa dell'equa-
zione lineare unidimensionale di Burgers.*/

calc_burgers(NX,F) :- (NX == 0;
                        NX == 1),
                        INF is 0.0,           /* Limite inferiore del dominio spaziale. */
                        SUP is 2.0 * pi,      /* Limite superiore del dominio spaziale. */
                        cond_iniziale_burg(NX,INF,SUP,F).

calc_burgers(NX,F) :- NX1 is NX - 1,
                        T is 0.6,             /* Tempo totale di simulazione. */
                        S is 0.1,              /* Costante di Courant-Friedrichs-Lewy (CFL). */
                        NU is 0.07,            /* Coefficiente di diffusione. */
                        INF is 0.0,            /* Limite inferiore del dominio spaziale. */
                        SUP is 2.0 * pi,        /* Limite superiore del dominio spaziale. */
                        DX is abs(SUP - INF) / NX1, /* Distanza tra qualsiasi coppia di punti
                                         della griglia adiacenti. */
                        DT is S * DX^2 / NU,   /* Lunghezza del passo temporale. */
                        NT is floor(T/DT),     /* Numero complessivo di passi temporali
                                         che deve effettuare l'algoritmo. */
                        cond_iniziale_burg(NX,INF,SUP,ONDA),
                        tempo_burg(0,NT,NX1,NU,DX,DT,ONDA,F).

/* Il predicato tempo_burg calcola numericamente l'integrazione della
funzione rispetto al parametro temporale DT:
- il primo argomento e' il numero di passi temporali che la funzione
d'onda ha compiuto;
- il secondo argomento e' il numero di passi temporali totali che la
funzione d'onda deve compiere;
- il terzo argomento e' il numero di passi spaziali utilizzati dal
predicato spazio_burg;
- il quarto argomento e' il coefficiente di diffusione;
- il quinto argomento e' la lunghezza del passo spaziale;
- il sesto argomento e' la lunghezza del passo temporale;
- il settimo argomento e' la funzione d'onda ricalcolata;
- l' ottavo argomento e' la funzione d'onda risultante. */

tempo_burg(NT,NT,_,_,_,_,F,F).
tempo_burg(I,NT,NX1,NU,DX,DT,ONDA,F) :- I < NT,

```

```

I1 is I + 1,
bordo_inf(ONDA,NU,DX,DT,BI),
spazio_burg(1,NX1,NU,DX,DT,ONDA,Z),
inserisci_elem(BI,Z,R),
tempo_burg(I1,NT,NX1,NU,DX,DT,R,F).

```

/* Il predicato spazio_burg calcola numericamente l'integrazione della funzione rispetto al parametro spaziale DX:

- il primo argomento e' l'indice per accedere agli elementi della lista, viene utilizzato da passo_eulero;
- il secondo argomento e' il numero di punti della funzione;
- il terzo argomento e' il coefficiente di diffusione;
- il quarto argomento e' la lunghezza del passo spaziale;
- il quinto argomento e' la lunghezza del passo temporale;
- il sesto argomento e' la funzione d'onda;
- il settimo argomento e' la funzione d'onda ricalcolata con il passo_eulero_burg e bordo_sup. */

```
spazio_burg(NX1,NX1,NU,DX,DT,ONDA,[F]) :- bordo_sup(ONDA,NU,DX,DT,F).
```

```
spazio_burg(I,NX1,NU,DX,DT,ONDA,[E|F]) :- I < NX1,
                                              I1 is I + 1,
                                              passo_eulero_burg(ONDA,I,NU,DX,DT,E),
                                              spazio_burg(I1,NX1,NU,DX,DT,ONDA,F).
```

/* Il predicato bordo_inf calcola le condizioni di bordo inferiore della funzione:

- il primo argomento e' la funzione d'onda;
 - il secondo argomento e' il coefficiente di diffusione;
 - il terzo argomento e' la lunghezza del passo spaziale;
 - il quarto argomento e' la lunghezza del passo temporale;
 - il quinto argomento e' il valore del punto estremo inferiore della funzione.
- L'uso di last e testa consente di selezionare gli elementi in coda e in testa alla funzione per il calcolo della condizione. */

```
bordo_inf([X|LX],NU,DX,DT,BI) :- last(LX,C),
                                         testa(LX,T),
                                         BI is X - X*DT/DX * (X - C) + NU*DT/DX^2 * (T - 2*X + C).
```

/* Il predicato bordo_sup calcola le condizioni di bordo superiore della funzione:

- il primo argomento e' la funzione d'onda;
 - il secondo argomento e' il coefficiente di diffusione;
 - il terzo argomento e' la lunghezza del passo spaziale;
 - il quarto argomento e' la lunghezza del passo temporale;
 - il quinto argomento e' il valore del punto estremo superiore della funzione.
- L'uso di last, testa e penultimo consente di selezionare l'ultimo, il primo e il penultimo elemento della funzione per il calcolo della condizione. */

```
bordo_sup(ONDA,NU,DX,DT,BS) :- last(ONDA,C),
                                         penultimo(ONDA,P),
                                         testa(ONDA,T),
                                         BS is C - C*DT/DX * (C - P) + NU*DT/DX^2 * (T - 2*C + P).
```

/* Il predicato passo_eulero_burg effettua il passo di Eulero:

- il primo argomento e' la funzione d'onda;
 - il secondo argomento e' l'indice dell'elemento corrente della lista;
 - il terzo argomento e' il coefficiente di diffusione;
 - il quarto argomento e' la lunghezza del passo spaziale;
 - il quinto argomento e' la lunghezza del passo temporale;
 - il sesto argomento e' il valore risultante del punto applicato il passo di Eulero.
 L'uso del predicato nth consente di accedere agli elementi della lista tramite
 indice. Gli elementi sono indicizzati da 1 a N. */

```

passo_eulero_burg(ONDA,I0,NU,DX,DT,EU) :- I1 is I0 + 1,
                                             I2 is I1 + 1,
                                             nth(I0,ONDA,E0),
                                             nth(I1,ONDA,E1),
                                             nth(I2,ONDA,E2),
                                             EU is E1 - E1*DT/DX * (E1 - E0) + NU*DT/DX^2 * (E2 - 2*E1 +
  
```

/* Il predicato cond_iniziale_burg calcola la condizione iniziale (una
 funzione) per il calcolo numerico dell'equazione di Burgers:
 - il primo argomento e' il numero di punti della griglia spaziale;
 - il secondo argomento e' l'estremo inferiore del dominio spaziale;
 - il terzo argomento e' l'estremo superiore del dominio spaziale;
 - il quarto argomento e' la funzione "onda a dente di sega". */

```

cond_iniziale_burg(NX,INF,SUP,ONDA) :- gen_punti_equi(NX,INF,SUP,L),
                                             onda_dente_sega(L,ONDA).
  
```

/* Il predicato onda_dente_sega calcola la funzione d'onda a dente di
 sega 'u':
 - il primo argomento e' la lista di punti equidistanti del dominio
 spaziale;
 - il secondo argomento e' la funzione d'onda calcolata.
 Per il calcolo dell'onda si fa uso dei predicati phi e phi_primo
 che sono rispettivamente la funzione e la sua derivata. */

```

onda_dente_sega([],[]).
onda_dente_sega([X|LX],[U|LU]) :- T0 is 0.0,
                                             NU is 0.07,
                                             phi_primo(X,T0,NU,F1),
                                             phi(X,T0,NU,F2),
                                             U is -2*NU*(F1 / F2)+4,
                                             onda_dente_sega(LX,LU).
  
```

```

phi_primo(X,T0,NU,F) :- F is -(-8*T0 + 2*X)*exp(-((-4*T0 + X)^2)/(4*NU*(T0 + 1)))/(4*NU*(T0 + 1)) - (-
  2*X - 4*pi)*exp(-((-4*T0 + X - 2*pi)^2)/(4*NU*(T0 + 1)))/(4*NU*(T0 + 1)).
  
```

```

phi(X,T0,NU,F) :- F is exp(-((X-4*T0)^2)/(4*NU*(T0+1))) + exp(-((X-4*T0-2*pi)^2)/(4*NU*(T0+1))).
```

/* Fine sezione equazione di Burgers */

/* Inizio sezione funzioni condivise */

```

/* Il predicato gen_punti_equi genera una lista di punti equidistanti tra loro:
 - il primo argomento e' il numero di punti che si vuole generare;
 - il secondo argomento e' l'estremo inferiore della lista di punti;
 - il terzo argomento e' l'estremo superiore della lista di punti;
 - il quarto argomento e' la lista di punti equidistanti.
 Per il calcolo dei punti si fa uso del predicato calc_punti. */

gen_punti_equi(0,_,_,[]).
gen_punti_equi(N,INF,SUP,L) :- N > 0,
                           DST is abs(SUP - INF),
                           N1 is N - 1,
                           ((SUP > INF,
                             calc_punti(0,N1,INF,DST,L));
                            (calc_punti(0,N1,SUP,DST,L1),
                             reverse(L1,L))).

calc_punti(N1,N1,INF,_,[INF]). 
calc_punti(I,N1,INF,DST,[INF|L]) :- I < N1,
                           I1 is I + 1,
                           INF1 is INF + (DST/N1),
                           calc_punti(I1,N1,INF1,DST,L).

/* Predicato che inserisce un elemento in testa alla lista. */

inserisci_elem(X,L,[X|L]). 

/* Predicato che estrae la lista di elementi successivi al primo. */

estrai_lista([_|LX],LX). 

/* Predicato per estrarre il primo elemento di una lista. */

testa([X|_],X). 

/* Predicato che restituisce il penultimo elemento di una lista. */

penultimo(LX,X) :- reverse(LX,LXInv),
                  estrai_lista(LXInv,Y),
                  testa(Y,X). 

/* Predicato per aggiungere un elemento alla fine di una lista. */

accoda_elem(X,[],[X]). 
accoda_elem(X,[Y|L],[Y|LX]) :- accoda_elem(X,L,LX). 

/* Fine sezione funzioni condivise */

```

5 Testing

5.1 Haskell

Test 1

```

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
| Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: -5
Acquisizione errata!
Il valore deve essere un numero naturale.
Digita il numero di punti totali della funzione d'onda: 100
[2.537825980164333, 2.5774635292715917, 2.61710107721347, 2.656738608716373, 2.6963759352420498, 2.736012966011584, 2.7756495287983616, 2.8152853722295843, 2.854920
131265235, 2.8945532832835124, 2.9341840924832963, 2.9738115409029056, 3.013434245078265, 3.0530503584023254, 3.09265746069832, 3.13225243816461, 3.1718313611743922,
3.211389362976298, 3.2509205391671836, 3.2904178727213695, 3.329873205461959, 3.3692772708399277, 3.4086198045315554, 3.447889746946439, 3.487075547488322, 3.52616557
3971027, 3.565148622103572, 3.604014509864235, 3.64275473064873, 3.6813631283736252, 3.7198365485475935, 3.7581754130997096, 3.796384164807898, 3.834471530536341, 3.87
24505616798967, 3.9103384250163393, 3.948155936600247, 3.9859268534865238, 4.023676960682313, 4.0614330108568115, 4.099221589534009, 4.137067986582411, 4.174995154672
836, 4.213022826959251, 4.2511668506558715, 4.289438772489547, 4.327845688844517, 4.3663963505872215, 4.4050714925905075, 4.443884342789181, 4.482821256305171, 4.52187
241704166, 4.561026551660163, 4.6002716079442605, 4.639595359793142, 4.678985912937524, 4.718432697474544, 4.757923744247111, 4.797451851014236, 4.837008650624908, 4.8
76587596546055, 4.916183280544297, 4.9557912919647045, 4.99540801442349, 5.035030329756987, 5.074655138113076, 5.114278485431834, 5.1538938248139, 5.193488374654201, 5
.233035309006789, 5.272476850138336, 5.31168754538108, 5.350394468348167, 5.3880040266101865, 5.423227236589644, 5.453274337330392, 5.472148504316077, 5.4671446638723
98, 5.412183765315732, 5.257413306490136, 4.922492502452091, 4.3295153627609938, 3.533568441198731, 2.803259325756947, 2.34624598099903, 2.134810078470153, 2.062664573
8837204, 2.055256271699049, 2.0755301911957758, 2.10725904901399, 2.1436722701026976, 2.181994724323144, 2.22109532101176004, 2.260513436500272, 2.3000613056481862, 2.3
39662707248342, 2.379284984717219, 2.418916610254246, 2.458551882606973, 2.498188637784255]
*Main> □

```

Test 2

```
*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----
| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale      = 100m,
| velocita' iniziale       = 10m/s.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.
-----
Digita lunghezza del passo temporale: 0
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 0.9
[100.0, 109.0, 118.0, 126.284851, 133.139404, 137.905336454689, 140.037978278134, 139.15862506155295, 135.09781356725566, 127.9254185669423, 117.96411620276123, 105.7838
3215383236, 92.17610146731589, 78.10878169402375, 64.66315672203721, 52.95702984786186, 44.058803877396706, 38.8986593581906, 38.18365822408268, 42.32383071871357, 51.
375993720634, 65.01158462948, 82.2065875052948, 164.147701459745, 139.4766407444373, 109.70833952320189, 76.80318495176581, 43.12659601347659, 11.29
2954, 196.12513263435542, 193.0219952745717, 182.28065875052948, 164.147701459745, 139.4766407444373, 109.70833952320189, 76.80318495176581, 43.12659601347659, 11.29
324919526108, -16.028880067833243, -36.2862822057439, -47.33254919258446, -47.5493719690744, -36.05900293417239, -12.844213318003415, 21.18196073031783, 64.174848813
10095, 113.43069711429229, 165.53324757539458, 216.56858177020052, 262.396578386679, 298.9619189271155, 322.62306495236817, 330.4744979377533, 320.6360795589583, 292.48
392709953146, 246.79981112227023, 185.82072981375313, 113.17678871364933, 33.713446601814894, 46.796936317994835, -122.04012341823137, 185.6186791617837, -231.55370
465824995, -251.6870749572778, -226.3887444846871, -161.14500730741088, -76.44286010463694, 29.010138523791176, 148.48344725899386, 273.59769001
264556, 394.8593895636503, 502.32684306856027, 586.36474619353, 638.4328128930749, 651.8541436499681, 622.4910646597652, 549.2771035619635, 434.5454797523207, 284.11
38484623635, 107.09889856518296, -84.5459218446654, -276.7548278334018, -454.29953032643925, -601.9069174450067, -705.469109584305, -753.2570761565056, -737.041661812
0278, -653.0255869398778, -592.49734457688226, -292.13293605902065, -33.89348602753546, 255.50523922739194, 555.5432747755533, 843.224708509467, 1094.7082180864448, 12
87.1343491005432, 1400.5199703972792, 1419.574709180137, 1335.2888305952565, 1146.148226044213, 858.85033572524, 488.4244612165678, 57.6995801808323, -403.88898696763
5, -764.0906529644918]
-----
| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale      = velocita' di trim,
| angolo iniziale          = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| velocita' di trim        = 30m/s,
| costante gravitazionale = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza = IN.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.
-----
Digita lunghezza del passo temporale: 0
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 2
[[10.0, 1000.0], [60.0, 1000.0], [119.019, 1000.0], [177.0753161528625, 998.7479065269398], [234.95290201868244, 995.0279962983323], [294.2079776547, 988.6508135495146], [1
356.640889322699095, 981.613820454273], [422.71217097668, 978.4372186345444], [489.33577382778975, 983.8075459069244], [550.2472491291718, 997.672602327661], [600.709
88374986, 1012.8544020596123], [640.8667355197858, 1019.0706715052474], [672.8843351677708, 1007.0996310307462], [696.138142753585, 965.5591221641714], [743.771178715
4609, 899.7622437167798], [848.5436839752847, 862.4564031301369], [952.4697900084046, 924.4426404787221], [951.7437040215162, 1021.357034896272], [993.4752938973656, 1
047.9695391385167], [876.9112199972899, 1024.6541628327059], [898.8852466786192, 967.8682229205278], [968.1145693846772, 900.6771074461807], [1089.375147235289, 899.3
725054324484], [1154.1925060735753, 997.5960151511746], [1114.1645149691585, 1068.1948751768948], [1069.042301398016, 1065.1185270097287], [1062.2427377384595, 1018.2
707222011862], [1102.5794411319405, 942.8144007206356], [1213.0310728695, 900.8018768563052], [1317.8934615980143, 974.7558516536983], [1298.5601345930577, 1074.10720
50306052], [1239.8640081869153, 1086.0853302988357], [1216.3965822557166, 1040.7130653834372], [1252.107941275579, 963.3310788512013], [1361.4869505574975, 916.775912
6051286], [1469.8636453073436, 989.277105801111], [1450.2399635573033, 1091.347757927018], [1388.750189669932, 1102.3234937802629], [1365.1339498602542, 1053.20241405
20907], [1404.5031456686006, 973.3178536840775], [1519.7331847816527, 932.9533996258536], [1620.7221053820408, 1016.4023678696908], [1589.1499806496877, 1112.69166757
63468], [1527.9257231590086, 1114.4527668843155], [1510.939231444575, 1057.848438070314], [1560.4428491210192, 975.9110604182742], [1684.8385184113813, 951.235348653
3109], [1767.4121512902025, 1051.7355085825511], [1718.6201225574137, 1133.414562605696], [1661.0051497710358, 1120.7576979457833], [1656.8968827343122, 1054.42470388
47125], [1724.0917334440987, 974.4954372420574], [1855.5381759650281, 977.7347589502833]]
```

```

| Calcolo dell'equazione di convezione lineare a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 0
[]

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda.
Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 0
[]

*Maint> □

```

Test 3

```

gabriele@gabriele-AMIGO-LI-2727:~/Documenti/III-anno/PLF/project/consegna/hs/src$ ghci
GHCi, version 8.6.5: http://www.haskell.org/ghc/ :? for help
Prelude :load simulazioni_numeriche.hs
[1 of 1] Compiling Main           ( simulazioni_numeriche.hs, interpreted )
Ok, one module loaded.
*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----
| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale      = 100m,
| velocita' iniziale       = 10m/s.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.
-----
Digita lunghezza del passo temporale: 1
[100.0, 110.0, 120.0, 129.019, 136.076, 140.2862361, 140.9574166, 137.675173385997, 130.37769550871997, 119.38280732793429, 105.40786721774319, 89.53147370868173, 73.12456842555967, 57.744625571615934, 45.00116255512479, 36.402951770085124, 33.26012693833372, 36.23617253796661, 45.825285684948966, 61.669630305956794, 82.82851440127145, 107.74760777635711, 134.35122386310638, 160.1947964103533, 182.66852035799354, 199.2371312301662, 207.6959602552197, 206.41962670659387, 194.578319456931, 172.29724682735127, 140.73804105904662, 102.0864753769788, 59.4385078670185, 16.585857122576584, -22.28771243619846, -52.79835219354104, -71.67256867046312, 75.35960879719883, -62.2055699373621, -31.848753454520164, 14.4204249439177, 73.6239750467626, 141.22287652636493, 211.40926605387983, 277.55169139415835, 332.7648677345513, 370.56022314917726, 385.52134583904375, 373.9405090379579, 334.35002908857797, 267.88598520255454, 178.4322034629416, 72.50880657495807, -41.10878947274003, -152.0294994454415, -249.10743717086717, -321.46128100069507, -359.56768524406084, -356.3287378212584, -308.06620047601365, -214.91781395050344, -81.80401915829628, 82.20321318245527, 264.0454198026357, 447.6334912096173, 615.1287069339603, 748.5210771706398, 831.3793212570979, 858.6176476731162, 798.1076626738131, 671.962086437774, 477.3321484934406, 226.59272986955784, -61.16297252153146, -361.3374217128244, -645.701783297551, -884.8089438166577, -1050.7627593918542, 1120.1068175786365, -1076.561049069078, -913.3228017550554, -634.663915527356, -256.59806244748586, 193.53832074561797, 678.6569738648202, 1154.5995177188772, 1573.7758124367954, 1889.4958944664918, 2060.6385692961385, 2056.231696878622, 1859.4861808131548, 1470.834352838947, 909.576895416864, 213.84060725848315, -561.3151743402921, -1347.6387195111245, -2069.0872460791743, -2648.522414263183, -3015.1701236068243, -3112.187784112473, 2903.607255489841, -2379.911105247121, -1561.561083240848]
-----
| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale      = velocita' di trim,
| angolo iniziale          = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| velocita' di trim        = 30m/s,
| costante gravitazionale = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza   = IN.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.
-----
Digita lunghezza del passo temporale: 2
[[0.0, 1000.0], [60.0, 1000.0], [119.019, 1000.0], [177.0753161528625, 998.7479065269398], [234.95290201868244, 995.0279962983323], [294.2079776547, 988.6508135495146], [356.64088932269095, 981.613820454273], [422.71217897668, 978.4372186345444], [489.33577382778975, 983.8075459069244], [550.2472491291718, 997.6722602327661], [608.70988374986, 1012.8544020596123], [640.8667355197858, 1019.0706715052474], [672.8843351677708, 1007.0996310307462], [696.138142753585, 965.559121641714], [743.7711787154689, 899.76224371677981], [848.5436839752847, 862.4564031301369], [952.4697900084046, 924.4426404787221], [951.7437048215162, 1021.357034896272], [983.4752938973656, 1047.9695391385167], [876.9112199972899, 1024.654162827059], [898.8852466786192, 967.6682229205278], [968.1145693846772, 900.6771074401807], [1089.375147235289, 899.3725054324484], [1154.1925060735753, 997.5960151511746], [1114.1645149691585, 1068.1948751768948], [1069.042301398016, 1065.1185270097287], [1062.2427377384595, 1018.2707222011862], [1102.5794411319405, 942.8144007206356], [1213.0310728695, 900.8018768563052], [1317.8934615980143, 974.7558516536983], [1298.5601345930577, 1074.1072050306052], [1239.8640081869153, 1086.0853302888357], [1216.3965822557166, 1040.7130053834372], [1252.107941275579, 963.3310788512013], [1361.4869505574975, 916.7759126051286], [1469.8636453073436, 989.27105801111], [1450.239963573033, 1091.347757927018], [1388.750189669932, 1102.3234937802629], [1365.1339498602542, 1053.2024140520907], [1404.5031456686006, 973.3178536840775], [1519.733187816527, 932.9533996258536], [1620.7221053820408, 1016.4023678696908], [1589.1490806496877, 1112.6916675763468], [1527.9257231590086, 1114.4527668843155], [1510.939231444575, 1057.8484380703314], [1560.4428491210192, 975.9110604182742], [1684.8385184113813, 951.2353486533109], [1767.4121512902025, 1051.7355085825511], [1718.6201225574137, 1133.414562605696], [1661.0051497710358, 1120.7576979457833], [1656.8968827343122, 1054.4247038847125], [1724.091734440987, 974.4954372420574], [1855.5381759650281, 977.7347589502833]]

```

```

| Calcolo dell'equazione di convezione lineare a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permesso una simulazione piu' accurata.
Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 1
[1.0]

| Calcolo dell'equazione di Burgers a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda.
Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 1
[4.0]
*Main> □

```

Test 4

```

*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele

| Calcolo del moto fugoide senza attrito
Parametri iniziali:
altitudine iniziale = 100m,
velocita' iniziale = 10m/s.
Parametri di simulazione:
secondi di simulazione = 100s,
costante gravitazionale = 9.81m/(s^2).
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permesso una simulazione piu' accurata.
Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 12
[100.0, 220.0, 340.0, -1235.168000000001, -6200.672000000004, 7694.941235200004, 110596.36741120002, 106208.61572227065, -1459095.0205642348, -4523331.405989825, 1.30
2560474083195e7]

| Calcolo del moto fugoide con attrito
Parametri iniziali:
velocita' iniziale = velocita' di trim,
angolo iniziale = 0rad,
spostamento laterale iniziale = 0m,
spostamento verticale iniziale = 1000m.
Parametri di simulazione:
secondi di simulazione = 100s,
velocita' di trim = 30m/s,
costante gravitazionale = 9.81m/(s^2),
coefficiente di resistenza dell'aria = 0.025,
coefficiente di portanza = 1N.
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permesso una simulazione piu' accurata.
Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 1000
[[0.0,1000.0],[30000.0,1000.0],[-185250.0,1000.0]]

```

```

| Calcolo dell'equazione di convezione lineare a una dimensione
| Parametri iniziali:
| estremo superiore del dominio spaziale = 2.0,
| estremo inferiore del dominio spaziale = 0.0,
| valore della parte alta della funzione = 2.0,
| valore della parte bassa della funzione = 1.0.
| Parametri di simulazione:
| numero di passi temporali da effettuare = 25,
| velocita' dell'onda = 1.0.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda,
| un valore alto permette una simulazione piu' accurata.
| Il valore del numero di punti dell'onda deve essere un naturale.
| Passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 21
Digita lunghezza del passo temporale: 0.3
[1.0,1.0,1.0,1.0,-3.355443099999814e7,1.2247367689999921e9,-2.1424504830999878e10,2.3904177356899863e11,-1.9098050232309897e12,1.1627929796608943e13,-5.6
060710748158734e13,2.1952763265024e14,-7.105646238105556e14,1.92450604760678318e15,-4.398132035911658e15,8.52538300858364e15,-1.4046770681118654e16,1.963740326
289396e16,-2.3095034961592216e16,2.2298447915458464e16]

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
| estremo superiore del dominio spaziale = 2.0 * pi,
| estremo inferiore del dominio spaziale = 0.0.
| Parametri di simulazione:
| tempo finale di simulazione = 0.6s,
| coefficiente di diffusione = 1.0m^2/s,
| costante di Courant-Friedrichs-Lowy = 0.1.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 35
[2.6001482895404073,2.8221091064914816,2.962953297613654,2.9749647809670834,3.2484421169296973,2.8159437723321434,3.24132138550074,2.107361
386453394,2.5450870348641375,0.3044484326650155,4.601443650646277,3.86576717322105,3.965957770973071,4.123027096713016,-18.960128327971468,-1248433,8128351818
,-3.804357859364016e14,-3.0124628025220063e30,-6.395918102872413e37,1.2879266971536558e37,-6.0191394754733475e34,3.651950130449536e29,-5.681241870769029e20,7.
947504686262198e9,308.23526035403484,2.30693329042999,1.7591741760237762,1.808382513565976,1.9125403480077081,2.0264424975757325,2.1412850466745192,2.256133
258059738,2.3709518498734257,2.485904035864941]

*Main> []

```

Test 5

```

*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele

| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale = 100m,
| velocita' iniziale = 10m/s.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 10
[100.0,200.0,300.0,-581.000000000001,-3424.0,413.610000000015,38821.66,74153.1958999999,-270374.7528000001,-1341364.553279,241002.9712100008,1.498313776336
5993e7,2.7362014407951877e7]

| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale = velocita' di trim,
| angolo iniziale = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| velocita' di trim = 30m/s,
| costante gravitazionale = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza = IN.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 0
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: -0.12
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 8
[[0.0,1000.0],[240.0,1000.0],[464.304,1000.0],[661.8291605053547,926.9684812146279],[934.9880535512357,612.9035274262772],[175.9321251459262,979.0533385465704
],[552.3135257720041,972.5694455565665],[290.95541475897693,1203.3063985087913],[262.7914313284634,1107.348228645912],[532.507493990252,455.8092136782715],[-8
54.1058994634004,146.19756327513244],[500.6672714567444,1760.7291236185406],[-2209.030853254841,2601.577701321431],[-6668.691673528916,709.1348558007492],[-16
83.4944242471993,11056.201849079931]]

```

```

Calcolo dell'equazione di convezione lineare a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 50
Digita lunghezza del passo temporale: -2
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 0
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 2
[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,-1.0740659143269653e42,2.633699127422579e43,-3.0944845928554555e44,2.318539060025998e45,-1.24365158661086
05e46,5.082578212969349e46,-1.644417596615775e47,4.320287207184026647,-9.379894139943642e47,1.703735368828876e48,-2.6110817763020997e48,3.39533981772282e48
3732152412968e47,9.204527913408536e47,-1.700322515826971e48,2.6106062796725456e48,-3.395297572195464e48,3.758139657260973e48,-3.5443706776368296e48,2.84532743
9055659e48,-1.9380715677460786e48,1.1138366527394053e48,-5.355402164200293e47,2.127881779208254e47,-6.865550547710572e46,1.753662256351069e46,-3.4128530019169
08e45,4.754966295450333e44,-4.2245576818105723e43,1.7984650426474192e42]

Calcolo dell'equazione di Burgers a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda.
Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 0
[]
*Main> []

```



```

| Calcolo dell'equazione di Burgers a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 70
[2.5603251504886444, 2.617313087551845, 2.6743003361134736, 2.7312862727233584, 2.788269923495047, 2.8452497337409888, 2.902223270300385, 2.9591868278009406, 3.016134923542116, 3.1299501094113285, 3.1867913561671, 3.243563958494142, 3.300243264764144, 3.356799148438868, 3.4131962073448157, 3.4693946363673285, 3.525351939200582, 3.581025579381527, 3.6363765584417296, 3.6913737523816272, 3.7459986515297796, 3.8002499621472, 3.8541473830426445, 3.9077338162177715, 3.9610753513724095, 4.014258602704632, 4.06738535820285, 4.1205649666356585, 4.173905337511415, 4.227503751827029, 4.2814387841786585, 4.335764480210162, 4.390507546605556, 4.44566778818734, 4.501221447316082, 4.557126776745785, 4.613330672978982, 4.669775450274559, 4.726404737843244, 4.783167879284047, 4.840022496383366, 4.8969351283844125, 4.953879863126045, 5.010834317637192, 5.06777039785781, 5.124631772478481, 5.1812748045790835, 5.2373076720637135, 5.291646162919132, 5.3412847832476835, 5.37791978485361, 5.37881438511586, 5.284790132888671, 4.95752083948166, 4.183768027460122, 3.092543889197522, 2.3434691794784466, 2.053251159659588, 1.989942863379328, 2.0082195166430075, 2.0530064478134227, 2.1061652720691257, 2.1619520166584345, 2.218563790090801, 2.2754344970593556, 2.3323865913005015, 2.38936407698239, 2.4463493939949714, 2.5033370170326315]
*Main> ■

```

Test 7

```

*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele

| Calcolo del moto fugoide senza attrito
Parametri iniziali:
altitudine iniziale = 100m,
velocita' iniziale = 10m/s.
Parametri di simulazione:
secondi di simulazione = 100s,
costante gravitazionale = 9.81m/(s^2).
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 1765
[100.0, 17750.0, 35400.0]

| Calcolo del moto fugoide con attrito
Parametri iniziali:
velocita' iniziale = velocita' di trim,
angolo iniziale = 0rad,
spostamento laterale iniziale = 0m,
spostamento verticale iniziale = 1000m.
Parametri di simulazione:
secondi di simulazione = 100s,
velocita' di trim = 30m/s,
costante gravitazionale = 9.81m/(s^2),
coefficiente di resistenza dell'aria = 0.025,
coefficiente di portanza = 1N.
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: -129
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: -2
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 10
[[0.0, 1000.0], [300.0, 1000.0], [791.5624656516269, 864.9919414233477], [1170.5394792779377, 209.79622282680998], [2567.2963356473438, 598.9909814181715], [1978.0197754075464, 770.5610190412983], [1998.6431950100541, 534.5962026930658], [2885.893852961458, 1340.8395316462475], [3004.7805496807414, 1428.0870339061987], [3443.566209721155, 1436.3920967313945], [3800.120018664817, 1125.9184010947315], [4279.75478121953, 50.08289134676488]]]

| Calcolo dell'equazione di convezione lineare a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
| Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 5
Digita lunghezza del passo temporale: 6
[1.0, -1.0834705943388371e26, 2.8465727433083995e27, -3.5727666705883976e28, 2.8484450065442973e29]

```

```

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: -23
Acquisizione errata!
Il valore deve essere un numero naturale.
Digita il numero di punti totali della funzione d'onda: 123
[2.531443832593343, 2.563622653747175, 2.595801528604663, 2.6279804089739236, 2.660159262404468, 2.692338058796651, 2.7245167621563615, 2.75669532399439, 2.7888736765
561815, 2.8210517248167637, 2.853293364975465, 2.8854063294965955, 2.9175824561886268, 2.949757384106197, 2.981930672594896, 3.014101745166393, 3.046269857471097, 3.0
7843406108895, 3.1105931636991904, 3.1427456866347234, 3.174889821349276, 3.2070233868952855, 3.2391437911057084, 3.271247998739009, 3.3033325103326137, 3.3353933558
494044, 3.4674261073262737, 3.399425914576471, 3.4313875675008307, 3.46330558768655226, 3.495174350700358, 3.526988238834561, 3.558741822091949, 3.5904300630017243, 3.6
22048538573869, 3.653593670501672, 3.6850629528048886, 3.7164551646764314, 3.747770555548941, 3.7790109894973054, 3.810180037147181, 3.8412830953001428, 3.87232689745
53426, 3.9833203021507994, 3.9342732103195636, 3.9651967673269256, 3.9961029696935455, 4.027094320260359, 4.057913458472839, 4.088842784220867, 4.119804940983875, 4.15
08680247928561, 4.181864881616191, 4.21298217833891, 4.24416676711391, 4.2754233325807265, 4.306755195844454, 4.338163762035046, 4.36964892680123, 4.401209171335305, 4
.432841753847383, 4.464542924696, 4.4963081525744775, 4.528132350126408, 4.560010088885794, 4.591935795581304, 4.6239039236912, 4.6559909996762425, 4.687946221746394, 4
.7209241777720502, 4.752097845805936, 4.816326218080236, 4.848460990824979, 4.886060600802023, 4.912759135235289, 4.944918635282347, 4.97708394901995
65, 5.009251177993566, 5.0414220142943975, 5.073594658898648, 5.105768201479777, 5.137941521467913, 5.17011293052769, 5.202279499344106, 5.234435766436636, 5.266571182
848788, 5.298665235628516, 5.33067772511783, 5.36252880936944, 5.394067359325406, 5.4249879737161715, 5.454700849684399, 5.482053057042952, 5.5048830505841234, 5.5186
211886415, 5.515247312113355, 5.4792963723335, 5.383450695907589, 5.3818784401634952, 4.823957870556797, 4.269571332098305, 3.5816827766754763, 2.936201498561026, 2
.4818176684923263, 2.229932615680866, 2.115879614577367, 2.077563553964675, 2.076798049564493, 2.0937891450990331, 2.1190035147854576, 2.1479936953961047, 2.178712447
1284165, 2.21022218005219, 2.242094090996528, 2.274132016665058, 2.3062461322931007, 2.338395255884008, 2.370560482433002, 2.402733122571867, 2.4349091776161815, 2.467
08680516406, 2.4992651548358373]
*Main> □

```

Test 8

```

*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele

| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
altitudine iniziale = 100m,
velocita' iniziale = 10m/s.
Parametri di simulazione:
secondi di simulazione = 100s,
costante gravitazionale = 9.81m/(s^2).
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: -90
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 0.876
[100.0, 108.76, 117.52000000000001, 125.62055083014491, 132.402203320576, 137.25515136167104, 139.66887496426617, 139.2780462108894, 135.90096098898448, 129.5670407251
4614, 120.5305109954153, 109.26818669247713, 96.46033402964785, 82.95477611334546, 69.71568278445359, 57.7597468462083, 48.083601758128296, 41.58728542310482, 38.99921
403357401, 40.80842759116296, 47.209755037578745, 58.066999525944176, 72.8982177889316, 90.88624293031368, 110.91442995135973, 131.62867882787083, 151.5213036180996
169, 0.03293457309078, 182.66606314213965, 191.1024210037743, 193.3157118888526, 188.67085027367534, 177.00122053750115, 158.65648593792503, 134.51513136559163, 105.958
1408390999, 57.32689898801564, 43.199063840537335, 43.492091996074043, 11.93892891261287, -30.8576703556413, -41.34969561762037, -41.99809968264536, -31.99117723688975
4, -11.302535479216136, 19.3223474037066978, 58.32603903373422, 103.40311091839146, 151.61738131489977, 199.57546693172074, 243.64781747339887, 280.2241681285395, 305.
98677061192035, 318.18217240348963, 314.8789754643113, 295.1351150012905, 259.22386565478486, 208.62290750110117, 146.03565251579784, 75.2713100671445, 1.0414227742762
99, -42.17751046164928, 57.61928808367179, 168.11915069818548, 281.8094157431933, 390.37169935221783, 485.24744548591165, 558.2641304224127, 602.795473092704, 611.79703
0362584922, 57.5031172391011, 516.6813357586118, 413.4616399760547, 278.87434591033485, 120.6897894857212, -50.96035357349351, -224.16801541134453, -386.01144439015343
,-523.451639504022, -624.3050944873423, -678.2253684071593, -677.6202549660903, -618.430658287743, -500.702131061477884, -328.89044159697823, -111.8581446371326, 137.4
6084703393458, 402.72843203926993, 665.1759800036657, 904.8342570548075, 1101.9463205401994, 1238.4707946778217, 1299.5691650117328, 1274.9639257045949, 1160.05561675
68357, 956.6965103870855, 673.5368954749283, 325.885444275586515, 64.94158480495724, -472.7731765239387, -868.1880340905957, -1220.4847642857349, -1499.8967004916553,
-1679.90309851032, -1739.4699205945894, -1665.046383250722, -1452.1483125608293, -1106.3782815750433, -643.7631688299386]

| Calcolo del moto fugoide con attrito
| Parametri iniziali:
velocita' iniziale = velocita' di trim,
angolo iniziale = 0rad,
spostamento laterale iniziale = 0m,
spostamento verticale iniziale = 1000m.
Parametri di simulazione:
secondi di simulazione = 100s,
velocita' di trim = 30m/s,
costante gravitazionale = 9.81m/(s^2),
coefficiente di resistenza dell'aria = 0.025,
coefficiente di portanza = 1N.
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 50
[[0.0,1000.0],[1500.0,1000.0],[2386.875,1000.0],[2824.0242269064747,1511.0888136058693],[20555.28895425848,5086.597929937973]]

```

Test 9

*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele

Calcolo del moto fugoide senza attrito
Parametri iniziali:
altitudine iniziale = 100m,
velocita' iniziale = 10m/s.
Parametri di simulazione:
secondi di simulazione = 100s,
costante gravitazionale = 9.81m/(s^2).
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 5.012
[100.0, 150.12, 200.24, 126.84997934483201, -193.56008262067198, -580.136176179772, -243.29623087191226, 1769.5938574585052, 4628.4640847719875, 3372.9773324278162, -9041.940637920638, -2952.41124059264, -27474.52436086445, 47571.4595695716, 190568.9613245647, 216583.203862304, -226772.57250189965, -1203604.9047426018, -1621358.304
1061945, 927161.5935670398, 7471418.694585241, 1.1731130772257913e7]

Calcolo del moto fugoide con attrito
Parametri iniziali:
velocita' iniziale = velocita' di trim,
angolo iniziale = 0rad,
spostamento laterale iniziale = 0m,
spostamento verticale iniziale = 1000m.
Parametri di simulazione:
secondi di simulazione = 100s,
velocita' di trim = 30m/s,
costante gravitazionale = 9.81m/(s^2),
coefficiente di resistenza dell'aria = 0.025,
coefficiente di portanza = 1N.
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: -1.02
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 9
[[0.0, 1000.0], [270.0, 1000.0], [520.1347499999999, 1000.0], [729.9893224155278, 898.5616594264133], [1044.6897968458243, 430.41056032594815], [677.8181515204399, -645.6098793169588], [256.3792336971636, -2123.4507214966407], [-1048.7477796075964, -1101.9376484251725], [-1249.1380213440418], [-1347.98552626632, -1045.0938463583657], [-1364.767263894925, 67.52395542238264], [-1657.7770603561469, 654.0403644351672], [-1666.52449902634, 715.3483417788642], [-973.429896673571, 507.2207748976873]]

Test 10

```
*Main> main
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele

-----| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale      = 100m,
| velocita' iniziale        = 10m/s.
| Parametri di simulazione:
| secondi di simulazione  = 100s,
| costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

-----| Digita lunghezza del passo temporale: -12.0
| Acquisizione errata!
| Il valore deve essere maggiore di zero.
| Digita lunghezza del passo temporale: 0
| Acquisizione errata!
| Il valore deve essere maggiore di zero.
| Digita lunghezza del passo temporale: 123
[100.0,1330.0,2560.0]

-----| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale          = velocita' di trim,
| angolo iniziale              = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione     = 100s,
| velocita' di trim            = 30m/s,
| costante gravitazionale     = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza      = 1N.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

-----| Digita lunghezza del passo temporale: 23
[[0.0,1000.0],[690.0,1000.0],[1250.26279197684],[1405.2362518459954,-4577.067502621624]]
```

```
| Calcolo dell'equazione di convezione lineare a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.
```

```
Digita il numero di punti totali della funzione d'onda: 2
Digita lunghezza del passo temporale: 0.2
[1.0,1.0]
```

```
| Calcolo dell'equazione di Burgers a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.
```

```
Digita il numero di punti totali della funzione d'onda: 3
[4.0,4.0,4.0]
*Main> 
```

5.2 Prolog

Test 1

```

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
| Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: -5.
Acquisizione errata!
Il valore deve essere un numero naturale.
Digita il numero di punti totali della funzione d'onda: 100.
[2.537825980164333, 2.5774635292715917, 2.617101077221347, 2.6567386087163731, 2.6963759352420498, 2.7360129660115842, 2.7756495287983616, 2.8152853722295843, 2.854920131265235, 2.8945532832835124, 2.9341840924832963, 2.9738115409029056, 3.013434245078265, 3.0530503584023254, 3.0926574606983199, 3.1322524384164612, 3.1718313611743922, 3.211389362976298, 3.2509205391671836, 3.2904178727213695, 3.3298732054619591, 3.3692772708399277, 3.4086198045315554, 3.4478897469464389, 3.4870755474883222, 3.5261655739710269, 3.565148622103572, 3.64275473064873, 3.6813631283736252, 3.7198365485475935, 3.7581754130997096, 3.7963841648078982, 3.8344715305363408, 3.8724505616799067, 3.9103384250163393, 3.948155936600247, 3.9859268534865238, 4.0236769606823133, 4.0614330108568115, 4.0992215895340092, 4.1370679865824114, 4.1749951546728363, 4.2130228269592509, 4.2511668506558715, 4.2894387724895466, 4.3278456888445174, 4.3663903505872215, 4.4050714925965075, 4.4438843427891808, 4.4828212563051713, 4.52187241704166, 4.5610265516601629, 4.6002716079442605, 4.6395953597931419, 4.6789859129375238, 4.718432097474544, 4.757923744247111, 4.7974518510142357, 4.8370086506249077, 4.876587596546055, 4.9161832805442973, 4.9557912919047045, 4.9954080144234903, 5.0350303297569869, 5.0746551381130756, 5.1142784854318339, 5.1538938248138999, 5.1934883746542013, 5.2330353090067891, 5.2724768501383359, 5.3116875453810799, 5.350394468348167, 5.3880940266101865, 5.4232272365896437, 5.4532743373303916, 5.4721485043160767, 5.4671446638723982, 5.4121837653157323, 5.2574133064901361, 4.9224925024520907, 4.3295153627609384, 3.5335684411987311, 2.8032259325756947, 2.3462459809990301, 2.1348100784701529, 2.0626645738837204, 2.055256271099049, 2.0755301911957758, 2.1072596490139902, 2.1436722701026976, 2.1819947243231441, 2.2210953210117004, 2.2605134365002719, 2.3000613056481862, 2.3396622707248342, 2.3792849847172191, 2.4189166102542461, 2.4585518826069732, 2.498188637784255]

true ?

(59 ms) yes
| ?- []

```

Test 2

```
| ?- main.
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----
| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale      = 100m,
| velocita' iniziale       = 10m/s.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 0.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 0.9.
[100.0, 109.0, 118.0, 126.284851, 133.13940400000001, 137.90533645468901, 140.03797872813399, 139.15862506155295, 135.09781356725566, 127.9254185669423, 117.96411620276
123.105.78383215383236, 92.176101467315888, 78.108781694023747, 64.663156722037215, 52.957029847861861, 44.058803877396706, 38.898659358190599, 38.183658224082677, 42
.32383071871573, 51.375991547200634, 82.51003771436325, 102.78916530315419, 124.45806278512413, 145.90533040294011, 165.40913589378735, 181.26525
792548657, 191.92390460992954, 196.12513263435542, 193.0219952745717, 182.28065875052948, 164.1477014597451, 139.47664074444373, 109.70833952320189, 76.803184951765
815, 125.6596013476593, 11.29349195735108, -16.020880067833243, -36.286282205743902, -47.332549192584459, -47.549371909074402, -36.059002934172391, -12.844213318003
415, 21.18196073017829, 64.174848813109951, 113.43069711429229, 165.5332477539458, 216.56858177020952, 262.39657838667898, 298.96191892711551, 322.62306495236817, 330
.47449793775331, 320.63607955895827, 292.48392709953146, 246.7989112227023, 185.82072981375313, 113.17678871364933, 33.713446601814894, 46.796936317994835, 122.040
123418231176, 148.48344725899386, 273.59769001264556, 394.85938956365032, 502.32684306856027, 586.36447461935302, 638.43281289307492, 651.85414364906842, 622.49106465
976524, 549.1270356196349, 434.545799752307, 284.11384846236348, 107.09889856518296, -84.54592184465401, -276.75482783340181, -454.29953032643925, -601.9069174450
0668, -705.46910958430499, -753.2570761565056, -737.9416181202777, -502.49734457688226, -292.13293605902065, -33.893486027535459, 255.5052392273
9194, 555.543274775553, 843.22470850946695, 1094.7082180864404, 1287.1343491065432, 1400.5199703972792, 1419.574709180137, 1355.2888305952565, 1146.1482260442131, 858
.8503572524003, 488.42446121656781, 57.699580180832299, 403.88989696763292, -862.11814645684719, -128.3067888431165, -1622.0445676705442, -1854.1017887497089, -194
9.3236264372047, -1889.2705818888596, 1666.376232660188, -1285.4124537240455, -764.0906529649182]

-----
```

```
| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale      = velocita' di trim,
| angolo iniziale          = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| velocita' di trim        = 30m/s,
| costante gravitazionale = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza   = IN.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 0.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 2.
[[10.0|1000.0], [60.0|1000.0], [119.01900000000001|1000.0], [177.07531615286251|998.747965269398], [234.9529020188244|995.02799629833225], [294.20797765470002|998
.65081354951462], [356.64088932269095|981.61382045427297], [422.71217097668|978.43721863454437], [489.33577382778975|983.80754590692436], [550.24724912917179|997.
67226023276612], [601.70988374985996|1012.8544020596123], [640.8667351978583|1019.0706715052474], [672.88433516777081|1007.0996310307462], [696.13814275358504|96
5.55912216417141], [743.77117871546091|899.76224371677984], [848.54368397528469|862.45640313013689], [952.46979000840463|924.44264047872297], [951.74370402151624|
1021.357034896272], [903.47529389736565|1047.9695391385167], [876.9112199728989|1024.6541623827059], [898.88524667861918|967.86822292052784], [968.11456938467722
|900.1074018075], [1089.375147235289|899.37250543244841], [1154.192506073573|997.59601515117458], [1114.1645149691585|1068.1948751768948], [1069.042391398016
|1065.1185270097287], [1062.2427377384595|1018.270722011862], [1102.5794411319405|942.8144007206356], [1213.0310728694999|900.80187685630517], [1317.893461598014
|3197.555813636983], [1298.561345930577|1074.107840081869153|1086.0853302988357], [1216.3965822557166|1040.7130653834372], [1252.10794127557
89|963.33107885120126], [1361.4869505574975|916.77591260512861], [1469.8636453073436|989.27716580111102], [1450.2399635573033|1091.3477579270179], [1388.750189669
9321|1102.3234937802629], [1365.1339498602542|1053.2024140520907], [1404.5031456686006|973.31785368407748], [1519.7331847816527|932.95339962585365], [1629.7221053
820408|1016.402367869698], [1589.1496806496877|1112.6916675763468], [1527.932731590886|1114.452766843155], [1510.9392314445749|1057.8484380703314], [1560.44284
91210192|975.91106041827425], [1684.8385184113813|951.23534865331089], [1767.412151290205|1051.7355085825511], [1718.6201225574137|1133.4145626056959], [1661.005
1497710358|1120.7576979457833], [1656.8968827343122|1054.4247038847125], [1724.0917334440987|974.49543724205739], 1855.538175965028332]
```

```
| Calcolo dell' equazione di concezione lineare a una dimensione
| Parametri iniziali:
| estremo superiore del dominio spaziale = 2.0,
| estremo inferiore del dominio spaziale = 0.0,
| valore della parte alta della funzione = 2.0,
| valore della parte bassa della funzione = 1.0.
| Parametri di simulazione:
| numero di passi temporali da effettuare = 25,
| velocita' dell'onda = 1.0.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda,
| un valore alto permette una simulazione piu' accurata.
| Il valore del numero di punti dell'onda deve essere un naturale.
| Passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 0.
[]
```

```

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
| Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 0.
[]
true ?

(5 ms) yes
|- □

```

Test 3

```

| ?- main.
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----
| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
altitudine iniziale = 100m,
velocita' iniziale = 10m/s.
| Parametri di simulazione:
secondi di simulazione = 100s,
costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 1.
[100.0, 110.0, 120.0, 129.0, 190.00000000001, 136.0759999999999, 140.2862361, 140.95741659999999, 137.67651733858997, 130.37769550871997, 119.38280732793429, 105.4078672174319, 89.53147307868173, 73.124568425559673, 57.744625571615934, 36.402951770958124, 33.20012693833717, 36.23617253796661, 45.825285684948966, 61.669630305956794, 82.82851440171121, 107.7476077635711, 134.35122386319638, 160.19479964103533, 182.66852035799354, 199.2371312301662, 207.69596025521969, 206.41962670659387, 194.57831945693101, 172.29724682735127, 140.73804105904662, 102.0864753769788, 59.438507867018501, 16.585857122576584, -22.28771243619846, -52.978352193541042, -71.672568670463122, -75.359608797198831, -62.2055699373621, -31.4848753454520164, 14.420429439176999, 73.623975046762595, 141.22287652636493, 211.40926605387983, 277.551619394158353, 332.76486773455127, 370.56022314917726, 385.52134503904375, 373.94050903797597, 334.35002968857797, 267.88598520255454, 178.4322034629416, 72.508806574958072, -81.088789472740028, -152.0294994454415, -249.10743717086717, -321.46128100669507, -359.56768524466084, -356.32873782125841, -308.00620047601365, -214.91781395050344, 81.804019158296285, 82.203213182455272, 264.04541980263571, 447.63349120961732, 615.128769339603, 748.52107717063984, 831.3793212570979, 850.61764767311615, 798.10766267381314, 671.96208643777743, 477.3321489344063, 226.59272986955784, 61.162972521531458, -361.33742171282438, -645.70178329975511, -884.80894381665769, -1120.1068175786365, -1076.5610490690781, 913.3228017550537, -634.66391552735604, -256.59806244748566, 193.53832074561797, 678.6569738648202, 1154.5995177188772, 1573.7758124367954, 1889.4958944664918, 2060.6385692961385, 2056.2316968786222, 1859.4861808131548, 1470.8343352838947, 909.57689541686398, 213.84060725848315, -561.31517430429206, -1347.6387195111245, -2069.0872460791743, -2648.5224142631828, -3015.1701236068243, -3112.1877841112473, -2903.607255489841, -2379.911052471212, -1561.5610832408479]

| Calcolo del moto fugoide con attrito
| Parametri iniziali:
velocita' iniziale = velocita' di trim,
angolo iniziale = 0rad,
spostamento laterale iniziale = 0m,
spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
secondi di simulazione = 100s,
velocita' di trim = 30m/s,
costante gravitazionale = 9.81m/(s^2),
coefficiente di resistenza dell'aria = 0.025,
coefficiente di portanza = IN.
| Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 2.
[[0.0|1000.0], [60.0|1000.0], [119.01900000000001|1000.0], [177.07531615286251|998.7479065269398], [234.95290201868244|995.02799629833225], [294.20797765470002|988.65081354951462], [356.64088932269095|981.61382045427287], [422.71217097668|978.43721863454437], [489.3357738277897|983.80754590692436], [556.24724912917179|997.67226023276612], [600.7098837498596|1012.854020596123], [640.86673551978583|1019.0706715052474], [762.88433516777081|1007.0996310307462], [696.13814275358504|965.55912216417141], [743.77117871546091|899.76224371677984], [848.54363897528469|862.45640313013689], [952.4697900840463|924.44264049782207], [951.74370402151624|1021.357034896272], [903.47529389736565|1047.9695391385167], [876.91121999728989|1024.654162387059], [898.88524667861918|967.86822292052784], [968.11456938467722|900.67710744018075], [1089.375147235289|899.37250543244841], [1154.1925060735753|997.59601515117458], [1114.1645149691585|1068.1948751768948], [1069.042301398016|1065.1185270097287], [1062.2427377384595|1018.270722011862], [1102.5794411319405|942.8144007206356], [1213.0310728694999|900.80187685638517], [1317.893461598014|31974.7558516536983], [1298.5601345930577|1074.1072050306052], [1239.8640081869153|1086.0853082988357], [1216.3965822557166|1040.7130653834372], [1252.10794127557|89|963.33107885120126], [1361.4869505574975|916.77591260512861], [1469.8636453073436|989.27710580111102], [1450.2399635573033|1091.3477579270179], [1388.750189669|9321.1102.3234937802629], [1365.1339498602542|1053.20241405209071], [1484.50314566860606|973.31785368407748], [1519.7331847816527|932.9533962585365], [1620.7221053|8204081916.4023678696908], [1589.1490806496877|1112.6916675763468], [1527.9257231590086|1114.452766843155], [1510.9392314445749|1057.8484380703314], [1560.44284|91210192|975.91106041827425], [1684.8385184113813|951.23534865331089], [1767.4121512902025|1051.7355085825511], [1718.6201225574137|1133.4145626056959], [1661.005|1497710358|[1120.7576979457833], [1656.896882733122|1054.4247038847125], [1724.0917334440987|974.49543724205739], [1855.538175965028332]
```

```

Calcolo dell' equazione di convezione lineare a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 1.
[1.0]

Calcolo dell'equazione di Burgers a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda.
Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 1.
[4.0]

true ?

(6 ms) yes
| ?- 
```

Test 4

```

| ?- main.
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----  

| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale = 100m,
| velocita' iniziale = 10m/s.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 12.
[100.0, 220.0, 340.0, -1235.1680000000001, -6200.6720000000005, 7694.9412352000036, 110596.36741120002, 106208.61572227065, -1459095.0205642348, -4523331.4059898248, 13
025604.747083195]  

-----  

| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale = velocita' di trim,
| angolo iniziale = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione = 100s,
| velocita' di trim = 30m/s,
| costante gravitazionale = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza = 1N.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 1000.
[[0.0|1000.0],[30000.0|1000.0],185250.0,1000.0]
```

```

Calcolo dell' equazione di convezione lineare a una dimensione
| Parametri iniziali:
| estremo superiore del dominio spaziale = 2.0,
| estremo inferiore del dominio spaziale = 0.0,
| valore della parte alta della funzione = 2.0,
| valore della parte bassa della funzione = 1.0.
| Parametri di simulazione:
| numero di passi temporali da effettuare = 25,
| velocita' dell'onda = 1.0.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda,
| un valore alto permette una simulazione piu' accurata.
| Il valore del numero di punti dell'onda deve essere un naturale.
| Passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 21.
Digita lunghezza del passo temporale: 0.3.
[1.0,1.0,1.0,1.0,1.0,-.33554430,999999814,1224736768,9999921,-21424504830,999878,239041773568,99863,-1909805023230,9897,11627929796608,943,-56060710748158.734,
219527632650240.0,-710564623810555.62,1924500476067831.8,-4398132035911658.0,8525383008583640.0,-14046770681118654.0,19637403262893996.0,-2390534961592216.0,
22298447915458464.0]

Calcolo dell' equazione di Burgers a una dimensione
| Parametri iniziali:
| estremo superiore del dominio spaziale = 2.0 * pi,
| estremo inferiore del dominio spaziale = 0.0.
| Parametri di simulazione:
| temp finale di simulazione = 0.6s,
| coefficiente di diffusione = 1.0m^2/s,
| costante di Courant-Friedrichs-Lowy = 0.1.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 35.
[2.6001482895440473,2.717221091064914816,2.9629532976136539,2.9749647809670834,3.248441169296973,2.8159437723321434,3.2413213855007399,2.107
36138645333942,2.5450870348641375,0.30444843266501548,4.601443650646277,3.8657671732210499,3.9659577709730711,4.1230270967130158,-18.960128327971468,-1284833.
128351818,-380435785936401.62,-3.0124628025220063e+30,-6.3959181028724134e+37,1.2879266971536558e+37,-6.0191394754733475e+34,3.6519501304495359e+29,5.6812418
707690291e+20,7947504686.2621984,-308.23526035403484,2.306933329042999,1.7591741/60237/62,1.8083825135659759,1.9125403480077081,2.0264424975757325,2.141285646
475192,2.25613325805097382,2.3709518498734257,2.4859040358649409]

true ?

(10 ms) yes
| ? - []

```

Test 5

```
| ?- main.
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----
| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale      = 100m,
| velocita' iniziale        = 10m/s.
| Parametri di simulazione:
| secondi di simulazione   = 100s,
| costante gravitazionale   = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 10.
[100.0,200.0,300.0,-.581.0000000000011,-3424.0,413.6100000000149,38821.660000000003,74153.195899999992,-270374.75280000007,-1341364.5532790001,241002.9712100
008,14983137.763365993,27362014.407951877]

-----
| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale          = velocita' di trim,
| angolo iniziale              = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione     = 100s,
| velocita' di trim            = 30m/s,
| costante gravitazionale     = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza       = 1N.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 0.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: -0.12.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 8.
[[0.0|1000.0],[240.0|1000.0],[464.3039999999997|1000.0],[661.82916050535471|926.9684812146279],[934.98805355123568|612.90352742627715],[175.93212514592619|97
9.05333854657044],[552.31352577200414|972.5694455656651],[290.99541475897693|1203.3063985087913],[262.79143132846337|1107.3482286459121],[532.50749399025199|
455.80921367840042|[146.19756327513244],[506.6672714567444|1760.7291236185406],[-2209.0308532548411|2601.5777013214311],[-6668.6916735289
16|709.1348558074916],[1683.494424247193,11056.201849079931]]
```

```

| Calcolo dell' equazione di convezione lineare a una dimensione
| Parametri iniziali:
| estremo superiore del dominio spaziale = 2.0,
| estremo inferiore del dominio spaziale = 0.0,
| valore della parte alta della funzione = 2.0,
| valore della parte bassa della funzione = 1.0.
| Parametri di simulazione:
| numero di passi temporali da effettuare = 25,
| velocita' dell'onda = 1.0.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda,
| un valore alto permette una simulazione piu' accurata.
| Il valore del numero di punti dell'onda deve essere un naturale.
| Passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 50.
Digita lunghezza del passo temporale: -2.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 0.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 2.
[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,-1.0740659143269653e+42,2.6336991274225788e+43,-3.0944845928554555e+44,2.3185390600259979e+45,-1.24365158
661086605e+46,5.0825782129693485e+46,-1.6444175966157751e+47,4.3202872071840257e+47,-9.3798941390436422e+47,1.7037353688288876e+48,-2.6110817763020907e+48,3.39
53398177722821e+48,-3.758140381660101e+48,3.5443443406455548e+48,-2.8450179905963733e+48,1.9357530286860528e+48,-1.1014001368732967e+48,4.8471443429033562e+4
7,-4.8346418259247849e+46,-3.6337321524129678e+47,9.2045279134085364e+47,-1.700322515826971e+48,2.6106062796725456e+48,-3.3952975721954643e+48,3.7581396572609
73e+48,-3.5443706776368296e+48,2.8453274390556588e+48,1.9380715677460786e+48,1.1138366527394053e+48,-5.3554021642002932e+47,2.1278817792082539e+47,-6.8655505
47710572e+46,1.7536622563510691e+46,-3.4128530019169079e+45,4.7549662954503328e+44,-4.2245576818105723e+43,1.7984650426474192e+42]

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
| estremo superiore del dominio spaziale = 2.0 * pi,
| estremo inferiore del dominio spaziale = 0.0.
| Parametri di simulazione:
| tempo finale di simulazione = 0.6s,
| coefficiente di diffusione = 1.0m^2/s,
| costante di Courant-Friedrichs-Lowy = 0.1.
| Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 0.
[]

true ?

(4 ms) yes
| ?- 
```

Test 6

```

| Calcolo dell'equazione di Burgers a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
| numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 70.
[2.5603251504886444, 2.6173130875518451, 2.6743003361134736, 2.7312862727233584, 2.7882699234950472, 2.8452497337409888, 2.902223270300385, 2.9591868278009406, 3.0161349235423116, 3.0730596794931286, 3.1299501094113285, 3.1867913561671002, 3.2435639584941418, 3.30024326476144, 3.356799148438868, 3.4131962073448157, 3.4693946363673285, 3.5810255793815271, 3.6363765584417296, 3.6913737523816272, 3.7459986515297796, 3.8002499621472001, 3.8541473830426445, 3.9077338162177715, 3.9610753513724095, 4.0142586027046319, 4.0673853582028503, 4.120564966356585, 4.1739053375114148, 4.2275037518270286, 4.281437841786585, 4.3357644802101616, 4.390507546677838718734, 4.456677838718734, 4.5012214473160821, 4.5571267767457853, 4.6133306729789822, 4.6697754562745594, 4.7264047378432439, 4.7831678792840471, 4.8400224963833658, 4.8969351283844125, 4.953879863126045, 5.0108343176371921, 5.0677703978578714, 5.1246317724784811, 5.1812748045790835, 5.2373076720637135, 5.2916461629191316, 5.3412847832476835, 5.377917984853612, 5.37881438511586, 5.2847901328886708, 4.9575208394816599, 4.1837680274601219, 3.0925438891975219, 2.3434691794784466, 2.0532515015, 2.3893640769823898, 2.4463493939949714, 2.5033370170326315]

true ?

(25 ms) yes
| ?- □

```

Test 7

```

| ?- main.
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele

| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale = 100m,
| velocita' iniziale = 10m/s.
Parametri di simulazione:
secondi di simulazione = 100s,
costante gravitazionale = 9.81m/(s^2).
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 1765.
[100.0, 17750.0, 35400.0]

| Calcolo del moto fugoide con attrito
Parametri iniziali:
velocita' iniziale = velocita' di trim,
angolo iniziale = 0rad,
spostamento laterale iniziale = 0m,
spostamento verticale iniziale = 1000m.
Parametri di simulazione:
secondi di simulazione = 100s,
velocita' di trim = 30m/s,
costante gravitazionale = 9.81m/(s^2),
coefficiente di resistenza dell'aria = 0.025,
coefficiente di portanza = 1N.
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: -129.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: -2.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 10.
[10.0, 1000.0, [300.0|10000.0], [575.4750000000002|1000.0], [791.56246565162689|864.99194142334773], [1170.5394792779377|209.79622282680998], [2567.2963356473438|598.69008141817149], [1978.0197754075464|770.56101904129832], [1998.6431950100541|534.59620269306583], [2885.8938529614579|1340.8395316462475], [3004.7805496807414|1428.087339061987], [3443.5662097211548|1436.3920967313945], [3800.1200186648171|1125.9184010947315], 4279.7547812195298, 50.082891346764882]

| Calcolo dell' equazione di convezione lineare a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 5.
Digita lunghezza del passo temporale: 6.
[1.0,-1.0834705943388371e+26, 2.8465727433083995e+27, -3.5727666705883976e+28, 2.8484450065442973e+29]

```



```

| Calcolo dell' equazione di convezione lineare a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita il numero di punti totali della funzione d'onda: 12.
Digita lunghezza del passo temporale: 0.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: -0.1.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 0.1.
[1.0,1.0,1.0,1.0,1.000000021395026,1.00000067513197,1.000010263273822,1.0000100086319104,1.0000703386875331,1.0003794097438439,1.0016335099573894,1.00576334775
71737,1.016976459266707]

-----
```

```

| Calcolo dell'equazione di Burgers a una dimensione
| Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

Digita il numero di punti totali della funzione d'onda: 21.
[1.963029017968277, 2.8854314654154951, -3.7591703623646291, 5.812804682399631, 3.2088548302682662, 3.4006298114687006, 3.592090666140296, 3.8012007651840292, 3.4937
057770489943, 0.10268289786064422, -700.53639712815118, -1789.5025156492904, -573.27628160351139, 13.93216171688821, 1.9874672625953997, 1.482881961903656, 1.67465497
68834542, 1.8663985426277945, 2.0590704436075975, 2.237328703452695, 2.5356259426799275]

true ?

(2 ms) yes
| ?- □

```

Test 9

```

| ?- main.
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----
```

```

| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
altitudine iniziale = 100m,
velocita' iniziale = 10m/s.
Parametri di simulazione:
secondi di simulazione = 100s,
costante gravitazionale = 9.81m/(s^2).
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 5.012.
[1.963029017968277, -193.56008262067198, -580.13617617977195, -243.29623087191226, 1769.5938574585052, 4628.4640847719875, 3372.977
322478162, -9041.9406379206375, -29522.411240592639, -27474.524360864449, 47571.459569571598, 190568.96132456471, 216583.20386230401, -226772.57250189965, -1203604.9
047426018, -1621358.3041061945, 927161.59356703982, 7471418.6945852414, 11731130.772257913]

-----
```

```

| Calcolo del moto fugoide con attrito
| Parametri iniziali:
velocita' iniziale = velocita' di trim,
angolo iniziale = 0rad,
spostamento laterale iniziale = 0m,
spostamento verticale iniziale = 100m.
Parametri di simulazione:
secondi di simulazione = 100s,
velocita' di trim = 30m/s,
costante gravitazionale = 9.81m/(s^2),
coefficiente di resistenza dell'aria = 0.025,
coefficiente di portanza = 1N.
Parametro richiesto:
passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: -1.02.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 9.
[[0.0|1000.0],[270.0|1000.0],[520.1347499999994|1000.0],[729.98932241552779|898.56165042641328],[1044.6897968458243|430.41056032594815],[677.81815152043987|-645.60987931695081],[256.37923369771636|-2123.4507214966407],[-1048.7477796075964|-1101.9376484251725],[-1249.6420682965511|-734.13802134404182],[-1347.98552626632|-1045.0938463583657],[-1364.7672638949225|67.523955422382642],[-1657.7770603561469|654.04036443516725],[-1666.52449902634|715.3483417780642],-973.42989667357097,507.22077489768731]
```

Test 10

```
| ?- main.
Progetto della sessione estiva del corso Programmazione Logica e Funzionale
Anno 2023/2024
Corso tenuto dal prof. Marco Bernardo
Progetto realizzato da: Barzotti Nicolas e Ramagnano Gabriele
-----
| Calcolo del moto fugoide senza attrito
| Parametri iniziali:
| altitudine iniziale      = 100m,
| velocita' iniziale        = 10m/s.
| Parametri di simulazione:
| secondi di simulazione  = 100s,
| costante gravitazionale = 9.81m/(s^2).
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: -12.0.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 0.
Acquisizione errata!
Il valore deve essere maggiore di zero.
Digita lunghezza del passo temporale: 123.
[100.0,1330.0,2560.0]

-----
| Calcolo del moto fugoide con attrito
| Parametri iniziali:
| velocita' iniziale          = velocita' di trim,
| angolo iniziale              = 0rad,
| spostamento laterale iniziale = 0m,
| spostamento verticale iniziale = 1000m.
| Parametri di simulazione:
| secondi di simulazione     = 100s,
| velocita' di trim            = 30m/s,
| costante gravitazionale     = 9.81m/(s^2),
| coefficiente di resistenza dell'aria = 0.025,
| coefficiente di portanza     = 1N.
| Parametro richiesto:
| passo temporale, determina la distanza temporale
| tra due punti di simulazione, un valore basso
| permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

Digita lunghezza del passo temporale: 23.
[[0.0 1000.0],[690.0 1000.0],[1250.262791976839],[1000.0],[1775.58369329037691|1006.7142233055527],[1090.892110923364|1133.6971520682619],[2389.7869961014545|148.58202791976839],-405.23625184599541,-4577.067502621624]
```

```

| Calcolo dell' equazione di convezione lineare a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0,
estremo inferiore del dominio spaziale = 0.0,
valore della parte alta della funzione = 2.0,
valore della parte bassa della funzione = 1.0.
Parametri di simulazione:
numero di passi temporali da effettuare = 25,
velocita' dell'onda = 1.0.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda,
un valore alto permette una simulazione piu' accurata.
Il valore del numero di punti dell'onda deve essere un naturale.
Passo temporale, determina la distanza temporale
tra due punti di simulazione, un valore basso
permette una simulazione piu' accurata.
| Il valore del passo temporale deve essere maggiore di zero.

```

Digita il numero di punti totali della funzione d'onda: 2.
 Digita lunghezza del passo temporale: 0.2.
 [1.0,1.0]

```

| Calcolo dell'equazione di Burgers a una dimensione
Parametri iniziali:
estremo superiore del dominio spaziale = 2.0 * pi,
estremo inferiore del dominio spaziale = 0.0.
Parametri di simulazione:
tempo finale di simulazione = 0.6s,
coefficiente di diffusione = 1.0m^2/s,
costante di Courant-Friedrichs-Lowy = 0.1.
Parametri richiesti all'utente:
numero di punti che compongono la funzione d'onda.
| Il valore del numero di punti dell'onda deve essere un naturale.

```

Digita il numero di punti totali della funzione d'onda: 3.
 [4.0,4.0,4.0]

true ?

(4 ms) yes
 | ?-