

Simulazione Numerica dell'Equazione di Burgers 1D: Analisi degli Shock e Stabilità Computazionale

Federico Spinello

9 gennaio 2026

Sommario

L'equazione di Burgers rappresenta un modello fondamentale per lo studio dei fenomeni non lineari in fluidodinamica, caratterizzato dalla competizione tra trasporto convettivo e diffusione viscosa. Questo lavoro presenta un'implementazione numerica completa per la risoluzione dell'equazione di Burgers unidimensionale con condizioni al contorno periodiche, utilizzando uno schema Runge-Kutta del quarto ordine per l'integrazione temporale e differenze finite ad alta precisione per le derivate spaziali. Particolare attenzione è dedicata all'analisi delle condizioni di stabilità attraverso i criteri di Courant-Friedrichs-Lewy (CFL), essenziali per prevenire l'insorgenza di instabilità numeriche che si manifestano come oscillazioni spurie, che esplodono rapidamente. Viene inoltre presentata un'analisi parametrica della dipendenza della pendenza degli shock dalla viscosità, verificando la relazione teorica $|\partial u / \partial x|_{\max} \propto \nu^{-1}$. I risultati mostrano buon accordo con la teoria e dimostrano l'efficacia dello schema numerico implementato per la simulazione di fenomeni fortemente non lineari.

1 Introduzione

L'equazione di Burgers, introdotta da Harry Bateman nel 1915 e successivamente sviluppata da Johannes Martinus Burgers nel 1948, costituisce un modello paradigmatico per lo studio di fenomeni non lineari in meccanica dei fluidi. Nonostante la sua apparente semplicità, questa equazione cattura caratteristiche fondamentali delle equazioni di Navier-Stokes, come la formazione di discontinuità (shock) e la cascata energetica verso scale più piccole. La forma dell'equazione è:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad (1)$$

dove $u(x, t)$ rappresenta la velocità del fluido, ν è il coefficiente di viscosità cinematica, e il termine $u \partial u / \partial x$ introduce la non linearità caratteristica del problema.

La fisica sottostante è governata dalla competizione tra due meccanismi: il termine convettivo non lineare tende a *steepening* (formazione di gradienti sempre più ripidi), mentre il termine diffusivo viscoso agisce come meccanismo di regolarizzazione. Per valori piccoli di ν , si osserva la formazione di shock, ovvero transizioni rapide della soluzione che in assenza di viscosità diventerebbero vere discontinuità.

Questo lavoro si concentra principalmente sugli aspetti computazionali della simulazione numerica dell'equazione di Burgers, con particolare enfasi sulla stabilità degli schemi numerici e sull'analisi quantitativa della struttura degli shock al variare della viscosità.

1.1 Teoria dello Scaling dello Shock

Dalla teoria degli shock per l'equazione di Burgers, in regime stazionario si ha un bilancio locale tra

convezione e diffusione:

$$u \frac{\partial u}{\partial x} \sim \nu \frac{\partial^2 u}{\partial x^2} \quad (2)$$

Introducendo lo spessore caratteristico dello shock δ e il salto di velocità Δu , fissato dalle condizioni di partenza ($\Delta u = 2$), otteniamo ch:

$$\frac{(\Delta u)^2}{\delta} \sim \nu \frac{\Delta u}{\delta^2} \quad (3)$$

il che equivale a dire che $R_e \sim 1$ sulla scala dissipativa da cui:

$$\delta \sim \frac{\nu}{\Delta u} \quad (4)$$

Il gradiente massimo scala quindi come:

$$\left| \frac{\partial u}{\partial x} \right|_{\max} \sim \frac{\Delta u}{\delta} \sim \frac{(\Delta u)^2}{\nu} \propto \nu^{-1} \quad (5)$$

Per quanto riguarda, invece, lo spettro di potenza per una discontinuità (shock), esso può essere derivato considerando la trasformata di Fourier di una funzione a gradino. Consideriamo uno shock ideale centrato in $x = x_0$ con salto Δu :

$$u(x) = \begin{cases} u_- & x < x_0 \\ u_+ & x > x_0 \end{cases} \quad (6)$$

La derivata spaziale di questa funzione è una delta di Dirac:

$$\frac{\partial u}{\partial x} = \Delta u \cdot \delta(x - x_0) \quad (7)$$

La trasformata di Fourier della derivata è legata alla trasformata di $u(x)$ dalla proprietà:

$$\mathcal{F} \left\{ \frac{\partial u}{\partial x} \right\} = ik \hat{u}(k) \quad (8)$$

Poiché la trasformata di Fourier della delta di Dirac è una costante:

$$\mathcal{F}\{\Delta u \cdot \delta(x - x_0)\} = \Delta u \cdot e^{-ikx_0} \quad (9)$$

si ottiene:

$$ik\hat{u}(k) = \Delta u \cdot e^{-ikx_0} \Rightarrow \hat{u}(k) = \frac{\Delta u}{ik} e^{-ikx_0} \quad (10)$$

Lo spettro di potenza (PSD), definito come $|\hat{u}(k)|^2$, diventa quindi:

$$\text{PSD}(k) = |\hat{u}(k)|^2 = \left| \frac{\Delta u}{ik} \right|^2 = \frac{(\Delta u)^2}{k^2} \quad (11)$$

dove il termine di fase e^{-ikx_0} scompare nel modulo quadro. Questa derivazione dimostra che **uno shock ideale produce uno spettro k^{-2}** .

Nel caso reale dell'equazione di Burgers, lo shock ha spessore finito $\delta \sim \nu/\Delta u$, introducendo un cutoff viscoso a $k_{\text{diss}} \sim 1/\delta$ che rappresenta il punto in cui il termine dissipativo diventa dello stesso ordine di grandezza del termine cinetico, questo si ottiene imponendo $R_e = 1$. Definiamo k_0 come il numero d'onda dell'onda di partenza (corrispondente a $k = 2\pi/L$). Lo spettro completo presenta quindi tre regimi:

$$\text{PSD}(k) \propto \begin{cases} k^0 & k < k_0 \quad (\text{armonica principale}) \\ k^{-2} & k_0 < k < k_{\text{diss}} \quad (\text{range inerziale}) \\ e^{-\nu k^2 t} & k > k_{\text{diss}} \quad (\text{range viscoso}) \end{cases} \quad (12)$$

La presenza dello spettro k^{-2} è quindi una *firma spettrale* degli shock e può essere utilizzata per diagnosticare la loro corretta formazione nelle simulazioni numeriche. Deviazioni significative da questa legge indicano insufficiente risoluzione spaziale o eccessiva dissipazione numerica.

2 Metodologia Numerica

2.1 Discretizzazione Spaziale

Il dominio spaziale $[0, L]$ viene discretizzato uniformemente con n_x celle, determinando un passo spaziale $\Delta x = L/n_x$. Vengono utilizzate condizioni al contorno periodiche, implementate mediante l'operatore modulo sugli indici, ad esempio, per calcolare la posizione subito a destra (o a sinistra) della cella i -esima, utilizziamo:

$$i_{\pm} = (i \pm 1 + n_x) \mod n_x \quad (13)$$

Per le derivate spaziali si adottano schemi alle differenze finite ad alta precisione per minimizzare l'errore numerico.

2.1.1 Derivata Prima

Viene utilizzato uno schema centrato a 5 punti, che fornisce accuratezza del quarto ordine ($\mathcal{O}(\Delta x^4)$):

$$\left. \frac{\partial u}{\partial x} \right|_i \approx \frac{u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}}{12\Delta x} \quad (14)$$

Questo schema presenta minimizza le oscillazioni numeriche rispetto agli schemi standard a 3 punti, mantenendo un costo computazionale contenuto.

2.1.2 Derivata Seconda

Per la derivata seconda si impiega uno schema centrato a 5 punti, anch'esso di quarto ordine:

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_i \approx \frac{-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}}{12\Delta x^2} \quad (15)$$

L'uso di schemi ad alta precisione è fondamentale per catturare accuratamente la fisica degli shock senza introdurre eccessiva dissipazione numerica.

È importante osservare che entrambi gli schemi mantengono lo stesso grado di errore ($\mathcal{O}(\Delta x^4)$), questo perchè l'errore complessivo rimarrebbe quello di ordine più basso.

2.2 Integrazione Temporale: Runge-Kutta 4

Per l'avanzamento temporale si utilizza il metodo esplicito di Runge-Kutta del quarto ordine (RK4), che garantisce accuratezza $\mathcal{O}(\Delta t^4)$ e buone proprietà di stabilità. Definendo l'operatore right-hand-side (RHS):

$$F(u) = \nu \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x} \quad (16)$$

lo schema RK4 si scrive:

$$k_1 = F(u^n) \quad (17)$$

$$k_2 = F(u^n + \frac{\Delta t}{2} k_1) \quad (18)$$

$$k_3 = F(u^n + \frac{\Delta t}{2} k_2) \quad (19)$$

$$k_4 = F(u^n + \Delta t k_3) \quad (20)$$

$$u^{n+1} = u^n + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (21)$$

Questo metodo richiede quattro valutazioni del RHS per timestep, ma garantisce elevata accuratezza e consente l'uso di timestep significativamente più grandi rispetto a metodi del primo ordine come Eulero esplicito.

2.3 Condizioni di Stabilità e Criteri CFL

La stabilità numerica rappresenta un aspetto critico nella simulazione di equazioni alle derivate parziali. Il timestep Δt deve soddisfare vincoli dettati sia dalla diffusione che dalla convezione, espressi attraverso i numeri di Courant-Friedrichs-Lewy (CFL).

2.3.1 Condizione CFL Diffusiva

Per il termine di diffusione $\nu \partial^2 u / \partial x^2$, la condizione di stabilità di von Neumann per schemi espliciti richiede:

$$CFL_{\text{diff}} = \frac{\nu \Delta t}{\Delta x^2} \leq C_{\text{diff}} \quad (22)$$

dove C_{diff} è il numero di Courant diffusivo critico, che dipende dallo schema di integrazione temporale. Per il metodo Runge-Kutta del quarto ordine combinato con differenze finite centrate, l'analisi di stabilità di von Neumann fornisce $C_{\text{diff}} \lesssim 0.5$. In questa analisi è stato scelto $C_{\text{diff}} = 0.4$. Questo valore è stato scelto conservativamente per garantire maggiore stabilità. Questo vincolo impone:

$$\Delta t \leq \frac{C_{\text{diff}} \Delta x^2}{\nu} \quad (23)$$

Si nota che per viscosità molto piccole, questa condizione diventa estremamente restrittiva, rendendo necessari timestep molto piccoli e aumentando significativamente il costo computazionale.

2.3.2 Condizione CFL Convettiva

Per il termine convettivo non lineare $u \partial u / \partial x$, la condizione di stabilità è:

$$CFL_{\text{adv}} = \frac{|u|_{\text{max}} \Delta t}{\Delta x} \leq C_{\text{adv}} \quad (24)$$

dove C_{adv} è il numero di Courant advettivo critico. Per lo schema RK4 con differenze finite centrate, l'analisi di stabilità indica $C_{\text{adv}} = 1$. Fisicamente, questa condizione assicura che l'informazione non "viaggi" numericamente per più di una cella durante un singolo timestep, prevenendo violazioni della causalità numerica. In questa analisi è stato utilizzato $CFL_{\text{adv}} = 0.3$, questa condizione è molto conservativa, ma garantisce stabilità nonostante non aumenti troppo il costo computazionale.

2.3.3 Time Step Adattivo

Nel codice implementato, il timestep viene ricalcolato ad ogni iterazione per garantire la stabilità:

$$\Delta t_{\text{diff}} = C_{\text{diff}} \frac{\Delta x^2}{\nu} \quad (25)$$

$$\Delta t_{\text{adv}} = C_{\text{adv}} \frac{\Delta x}{|u|_{\text{max}}} \quad (26)$$

$$\Delta t = s \cdot \min(\Delta t_{\text{diff}}, \Delta t_{\text{adv}}) \quad (27)$$

dove $s = 0.9$ è un fattore di sicurezza aggiuntivo. Questo approccio adattivo è essenziale poiché $|u|_{\text{max}}$ varia dinamicamente durante la simulazione, specialmente nella fase di formazione degli shock.

2.3.4 Instabilità Numerica

La violazione dei criteri CFL porta a instabilità numeriche. Imponendo $s = 2$ osserviamo una differenza con la simulazione eseguita con $s = 0.9$ sostanziale. Inizialmente le due simulazioni sono sovrapponibili, ma dopo un certo tempo iniziano ad essere amplificati gli errori numerici, che portano ad oscillazioni spurie, su quasi tutte le frequenze. Il codice, a questo punto, non sta più simulando nulla di fisico.

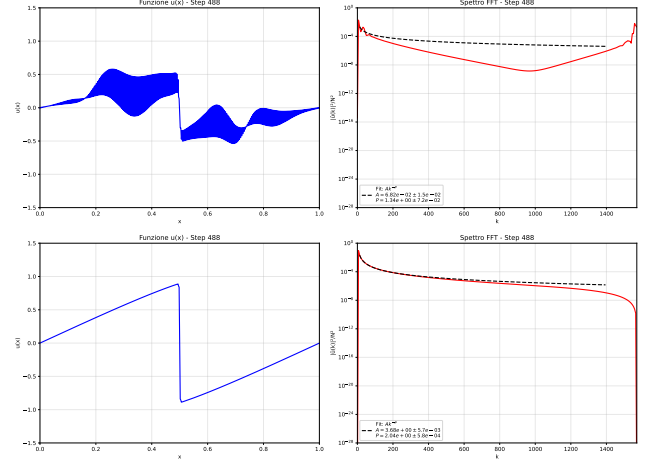


Figura 1: Instabilità numerica causata dalla violazione della condizione CFL. In alto simulazione con $s = 2$, in basso stessa simulazione allo stesso step temporale, ma con $s = 0.9$. Sono ben visibili le oscillazioni ad alta frequenza dovute all'errore numerico che viene amplificato ad ogni step temporale.

La Figura 1 mostra un esempio tipico di questo fenomeno: le oscillazioni iniziano come piccole perturbazioni ma rapidamente dominano la soluzione, invalidando completamente i risultati numerici.

3 Analisi Spettrale FFT

Per caratterizzare la distribuzione energetica alle diverse scale, si utilizza la trasformata di Fourier discreta implementata tramite la libreria FFTW3 (Fastest Fourier Transform in the West).

Data la natura reale del campo $u(x)$, si impiega la trasformata ottimizzata reale-complessa `fftw_plan_dft_r2c_1d` invece della trasformata complessa-complessa standard (`fftw_plan_dft_1d`). Quando i dati di input sono reali, l'output della trasformata di Fourier soddisfa la simmetria hermitiana: $\hat{u}(k) = \hat{u}^*(-k)$. Conseguentemente, solo i coefficienti per $k \geq 0$ sono indipendenti. La r2c sfrutta questa proprietà calcolando soltanto $N/2 + 1$ coefficienti complessi invece di N , dimezzando sia il costo computazionale che i requisiti di memoria rispetto alla FFT complessa standard.

La densità spettrale di potenza (PSD) è definita come:

$$PSD(k) = \frac{|\hat{u}(k)|^2}{N^2} \quad (28)$$

dove $\hat{u}(k)$ è la trasformata di Fourier di $u(x)$, N il numero di punti, e il numero d'onda è $k = 2\pi n/L$ con $n = 0, 1, \dots, N/2$.

L'analisi spettrale permette di osservare:

- La cascata energetica dai bassi k agli alti k
- La formazione dello shock, con $\text{PSD} \propto k^{-2}$
- La dissipazione dovuta alla viscosità, alimentata dalla cascata di energia
- La presenza di eventuali oscillazioni numeriche spurie (picchi ad alte frequenze non fisici)

4 Risultati: Simulazione principale

La simulazione principale (`sim_adv_conv`) utilizza una condizione iniziale sinusoidale:

$$u(x, 0) = \sin\left(\frac{2\pi x}{L}\right) \quad (29)$$

con $L = 1$, $n_x = 2048$, e viscosità $\nu = \Delta x^2 \pi \cdot 800/2 \approx 6 \times 10^{-4}$. I parametri sono scelti per garantire una risoluzione adeguata dello shock, mantenendo un costo computazionale ragionevole. In particolare, la viscosità è stata scelta in questo range poichè una viscosità più piccola avrebbe ridotto la dimensione dello shock. Affinchè il sistema non divergesse si sarebbe dovuto aumentare la risoluzione della griglia spaziale, in modo da permettere al programma di risolvere correttamente lo shock, ma questo avrebbe aumentato il costo computazionale. Al contrario, se la viscosità fosse troppo elevata, avrebbe dissipato l'energia molto più velocemente del tempo necessario alla formazione di shock.

4.1 Dinamica Temporale

La dinamica osservata segue tre fasi distinte:

1. **Fase iniziale:** Il termine convettivo domina, causando *steepening* del profilo. L'energia si trasferisce progressivamente dalla frequenza fondamentale verso frequenze più alte.
2. **Formazione shock:** I gradienti crescono fino a raggiungere uno spessore determinato dall'equilibrio locale tra convezione e diffusione. Questa fase è critica per la stabilità numerica poichè $|u|_{\max}$ e $|\partial u / \partial x|_{\max}$ raggiungono i loro valori massimi.
3. **Quasi stazionarietà:** La soluzione si sposta lentamente verso una soluzione quasi stazionaria.
4. **Cascata di energia e dissipazione:** La viscosità dissipa continuamente energia alle piccole scale, rimuovendo le alte frequenze generate dalla cascata. Ovvero la dissipazione inizia a dissipare lo shock, riducendone progressivamente ampiezza e gradiente.

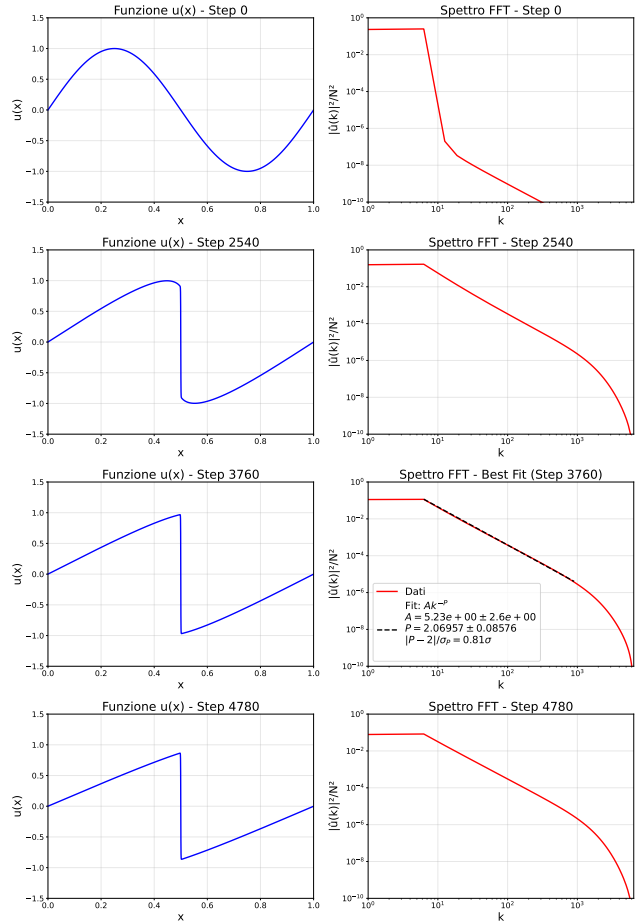


Figura 2: Simulazione principale: istantanee temporali rappresentative delle diverse fasi evolutive.

Nei grafici in Figura 2 è visibile l'evoluzione temporale della simulazione e i risultati del fit. Il fit è stato eseguito nel grafico 3 dove è rappresentato uno stato quasi stazionario. Nel grafico 1 si osserva la condizione iniziale, con la trasformata che mostra un picco per bassi $k \sim 1$ e l'errore numerico su piccola scala. Nel grafico 2 si vede la formazione di shock: a questo punto sono state generate quasi tutte le frequenze (entro la risoluzione numerica). Nel grafico 3 vediamo la soluzione quasi stazionaria, il termine convettivo bilancia il termine diffusivo. In questo punto è stato eseguito un fit con la legge di potenza Ak^{-P} . L'ultimo grafico, invece, mostra per quanto tempo la soluzione rimane quasi stazionaria ~ 1000 timesteps.

4.2 Analisi Quantitativa del Miglior Fit

Per identificare l'istante temporale in cui lo spettro segue più fedelmente la legge k^{-2} , è stata condotta un'analisi sistematica su tutti i timestep della simulazione. Per ogni istante temporale è stato calcolato il fit con la legge di potenza $\text{PSD}(k) = Ak^{-P}$ nel range $1 < k < 900$, e è stato valutato l'istante temporale in cui veniva minimizzato $p - 2$. Grazie a questa analisi è stato ottenuto il seguente risultato:

$$p_{\text{fit}} = 2.07 \pm 0.09 \rightarrow |\text{fit} - 2| \approx 0.81\sigma \quad (30)$$

Il fit è eseguito nel range inerziale $6 = k_0 < k < k_{\text{diss}} = 900$, escludendo il picco a bassi k e la regione dissipativa ad alti k . I valori entro il quale eseguire il fit sono stati ottenuti sperimentalmente, ovvero graficando i dati e vedendo l'andamento.

5 Analisi Parametrica: Shock vs Viscosità

La seconda modalità di simulazione (`sim.shock.analysis`) esegue una scansione parametrica sulla viscosità per quantificare la dipendenza della pendenza dello shock da ν .

5.1 Implementazione Computazionale

L'analisi procede come segue:

1. Per ogni valore di ν nel range $[\nu_{\text{min}}, \nu_{\text{max}}]$ (in scala logaritmica):
 - Inizializza $u(x, 0) = \sin(2\pi x/L)$
 - Evolvi fino a t_{final} per raggiungere lo stato stazionario
 - Calcola $\partial u / \partial x$ usando lo schema a 5 punti
 - Misura $|\partial u / \partial x|_{\text{max}}$ in $x = 0.5$ (posizione dello shock)
2. I risultati vengono salvati ordinati in `shock_values.txt`
3. L'analisi viene visualizzata in scala log-log per verificare la legge di potenza

5.2 Risultati Parametrici

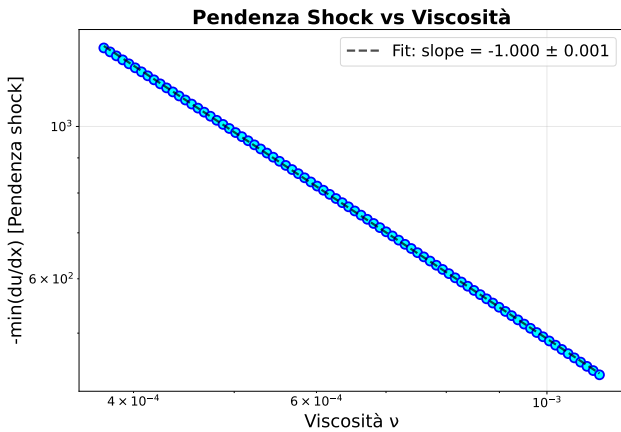


Figura 3: Pendenza massima dello shock in funzione della viscosità in scala log-log. La linea nera tratteggiata rappresenta il fit $|\partial u / \partial x|_{\text{max}} = A\nu^\alpha$ con $\alpha \approx -1.000 \pm 0.001$.

La Figura 3 mostra la legge di potenza aspettata. Un fit lineare in scala log-log fornisce un esponente $\alpha = -1.000 \pm 0.001$. Questo valore è in ottimo accordo con il valore teorico $\alpha = -1$.

6 Dettagli Implementativi

6.1 Architettura del Codice

Il codice è composto in modo da separare la parte di compilazione (C) e di analisi (py), in modo da permettere ricompilazioni parziali.

È possibile selezionare tra 2 simulazioni tramite il file `main.c`:

- `sim.adv.conv.c`: Simulazione principale dello shock. Fornisce la velocità e lo spettro in funzione del tempo.
- `sim.shock.analysis.c`: Scansione parametrica su ν .

Tutti i parametri sono configurabili tramite il file `params.txt`, permettendo rapide modifiche senza ricompilazione.

Inoltre, sono stati scritti anche 2 file che contengono le funzioni ausiliarie alla compilazione (`functions.c/h`, come Derivate, RHS, integratore RK4, FFT) e alla lettura dei parametri salvati sul file di testo `params.txt` (`params.c/h`)

Per quanto riguarda l'analisi dati è stato utilizzato python, che grafica i dati ottenuti dalla simulazione principale (`plot.py`) e dalla scansione parametrica (`plot.shock.py`). Il file `find_best_fit.py` esegue il fit dello spettro visibile in Figura 2 per ogni step temporale, e trova il punto in cui il risultato si avvicina di più al valore atteso e restituisce i risultati del fit.

Tutto quanto è gestito dal file Makefile, che permette di compilare il c/h, fornendo pratici comandi che eseguono le varie simulazioni in modo automatico.

7 Conclusioni

Questo lavoro ha presentato un'implementazione numerica completa e robusta per la simulazione dell'equazione di Burgers 1D, con particolare enfasi sugli aspetti computazionali di stabilità e accuratezza. I principali risultati sono:

1. **Stabilità numerica:** L'implementazione di criteri CFL adattivi è essenziale per prevenire instabilità. Il time step adattivo garantisce stabilità in tutte le fasi della simulazione, dal regime lineare iniziale alla formazione dello shock.
2. **Accuratezza:** L'uso di schemi alle differenze finite del quarto ordine combinati con RK4 fornisce eccellente accuratezza, minimizzando l'errore numerico, pur mantenendo un costo computazionale ragionevole. Questo è critico per catturare correttamente la fisica degli shock.
3. **Validazione quantitativa:** L'analisi parametrica conferma la legge di scaling teorica $|\partial u / \partial x|_{\text{max}} \propto \nu^{-1}$ con pendenza fittata $= -1.000 \pm 0.001$. Lo spettro k^{-2} è confermato con ottimo accordo ($P = 2.07 \pm 0.09$).

4. **Efficienza computazionale:** L'architettura modulare e le ottimizzazioni implementate rendono il codice sufficientemente veloce per analisi parametriche estese.