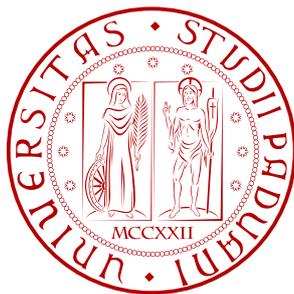


UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E
NATURALI
DIPARTIMENTO DI FISICA "Galileo Galilei"



LAUREA TRIENNALE IN FISICA

Modelli differenziali in fisica del traffico

Relatore: Prof. Francesco Fassò

Correlatore: Prof. Armando Bazzani (Università degli Studi di Bologna)

Laureanda: Elisa Omodei

ANNO ACCADEMICO 2008/2009

Indice

1	Introduzione	3
2	Modelli di traffico	4
2.1	Modelli di velocità ottimale	4
2.2	Altri modelli	5
3	Studio della stabilità di una soluzione di un semplice modello di velocità ottimale	7
3.1	Auto che si muovono lungo una circonferenza	7
3.2	Auto che si muovono lungo una retta	12
4	Stop-and-go	14
4.1	Simulazioni Numeriche	14
4.1.1	Auto che si muovono lungo una circonferenza	14
4.1.2	Auto che si muovono lungo una retta	15
5	Appendice	19
5.1	Codice del programma che simula il moto di n auto lungo una circonferenza	19
5.2	Codice del programma che simula il moto di n auto lungo una linea retta	21

1 Introduzione

Il traffico delle auto è un esempio di sistema a molti corpi e gli ingorghi stradali sono un'indicazione del comportamento complesso di tale sistema.

Sono stati creati diversi modelli per cercare di capire la ricca varietà di fenomeni che il traffico presenta e ciascuno di questi modelli mette in evidenza certi aspetti del problema. In particolare, nei modelli microscopici il traffico è trattato come un sistema di particelle lontano dall'equilibrio. Nei modelli macroscopici è invece visto come un fluido comprimibile.

Tipici dati sperimentali vengono forniti sotto forma di diagramma che rappresenta il flusso in funzione della densità (chiamato in letteratura diagramma fondamentale); si veda ad esempio la Figura 1. Su di esso si vede un tipico fenomeno, che è il passaggio dal traffico che scorre libero a basse densità al traffico congestionato ad alte densità. Questa tesi è dedicata ad uno dei possibili approcci allo studio di questo fenomeno, per la descrizione del quale sono adatti i modelli microscopici.

Nel capitolo 2 descriverò brevemente i diversi modelli sviluppati per la fisica del traffico, basandomi su [1]. L'argomento centrale della tesi è studiato nei capitoli successivi. Nel capitolo 3 studierò un particolare modello, "optimal velocity model"[3] [4], in due casi: nel caso in cui le auto si muovono lungo una circonferenza e nel caso in cui si muovono lungo una linea retta. Nel capitolo 4 presenterò delle simulazioni numeriche che mettono in evidenza il passaggio dal traffico libero a quello caotico.

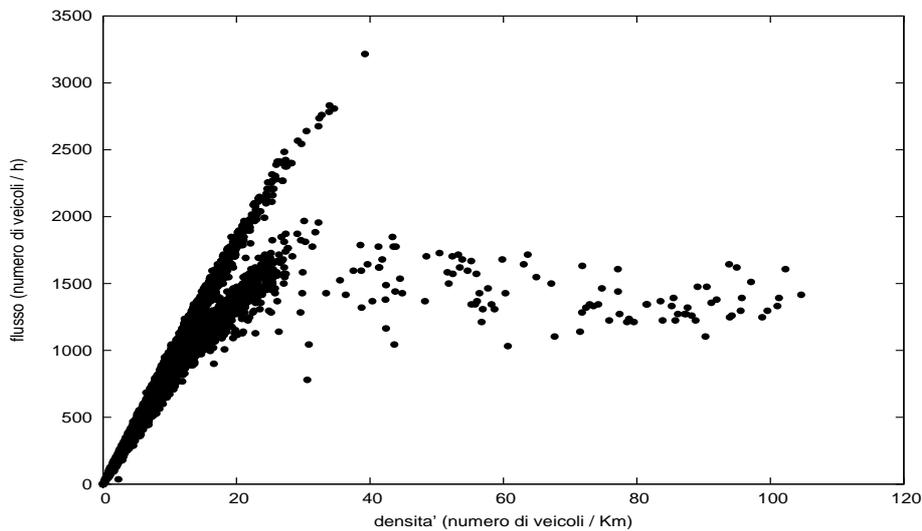


Figura 1: Flusso di auto sull'autostrada Firenze-Pisa-Livorno (2008)

2 Modelli di traffico

Cominciamo con una breve rassegna basata sull'articolo *The physics of traffic jams* di Nagatani [1], al quale si rimanda per la bibliografia.

2.1 Modelli di velocità ottimale

Ipotesi di base di questi modelli è che ogni conducente cerchi di mantenere una velocità "ottimale" che dipende dalla distanza dal veicolo davanti.

- Un primo modello è quello proposto da Newell [2], nel quale l'equazione del moto del veicolo j -esimo è:

$$\frac{dx_j}{dt}(t + \tau) = V(x_{j+1}(t) - x_j(t)) \quad (1)$$

dove $x_j(t)$ è la posizione del veicolo j -esimo al tempo t . La funzione reale V rappresenta la velocità ottimale. Il parametro positivo τ misura il ritardo con il quale il veicolo adegua la sua velocità in base alla distanza da quello davanti e per semplicità è supposto uguale per tutti i veicoli.

- Modello differenziale del II ordine: supponendo che τ sia piccolo, Bando *et al* [3] hanno espanso in serie di Taylor in τ al primo ordine la velocità nell'equazione (1), ottenendo l'equazione

$$\frac{d^2x_j}{dt^2}(t) = b \left(V(x_{j+1}(t) - x_j(t)) - \frac{dx_j}{dt}(t) \right) \quad (2)$$

ove il parametro $b = \frac{1}{\tau}$ è chiamato sensibilità. Questa equazione differenziale è del tipo di quella di un sistema di masse e molle con smorzamento e interazioni non lineari.

- Modelli a differenze finite: trasformando la derivata rispetto al tempo in una differenza finita nell'equazione (1), Nagatani ha ottenuto il modello:

$$x_{j+1}(t + 2\tau) - x_j(t + 2\tau) = x_{j+1}(t + \tau) - x_j(t + \tau) + \tau [x_{j+1}(t + \tau) - x_j(t + \tau) - V(x_{j+1}(t) - x_j(t))]$$

In questi modelli generalmente si assume che la velocità ottimale $V : \mathbb{R} \rightarrow \mathbb{R}$ sia una funzione monotona, crescente, differenziabile quasi ovunque e limitata con asintoto orizzontale v_∞ ; la funzione tende ad un valore massimo quando la distanza tra le auto tende all'infinito.

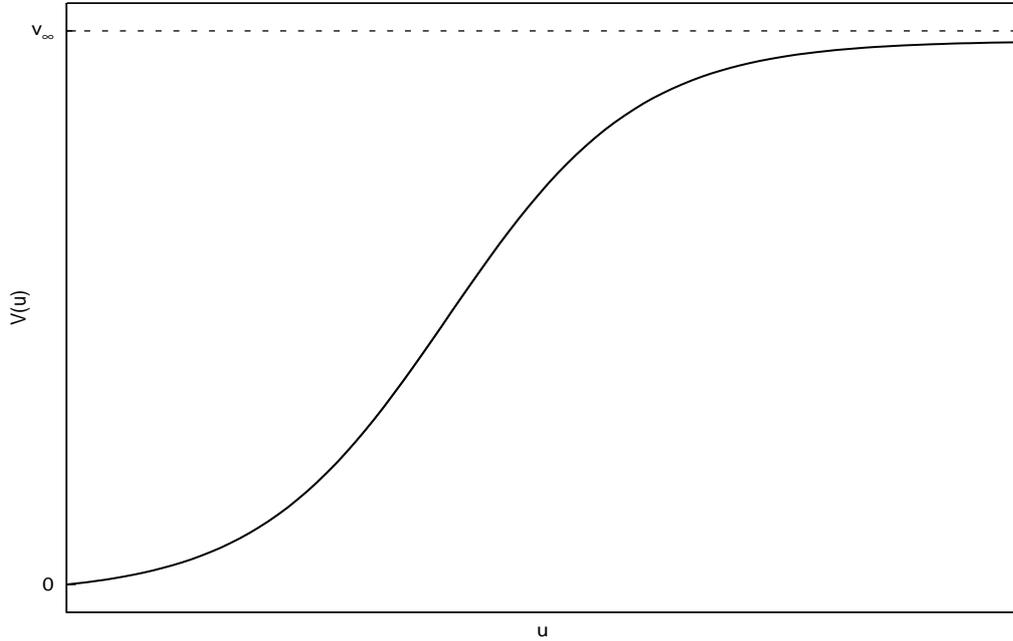


Figura 2: Grafico della funzione velocità ottimale (3)

Nel modello di Bando si utilizza la funzione

$$V(u) = \frac{v_\infty}{2} [\tanh(u - x_c) + \tanh(x_c)] \quad (3)$$

dove x_c è una costante positiva che rappresenta una "distanza di sicurezza" tra le auto [3]. Si veda la Figura 2.

Un'altra funzione è quella proposta da Bazzani [4], definita per $u > x_c$ da

$$V(u) = v_\infty \left(1 - \left(\frac{x_c}{u}\right)^a\right)^m \quad (4)$$

dove a e m sono due ulteriori parametri reali positivi; essa tende a zero per $u \rightarrow x_c$ (Figura 3).

I modelli di velocità ottimale sono semplici e convenienti per simulazioni numeriche e per l'analisi teorica. Esistono anche dei raffinamenti di questi modelli che tengono in considerazione ulteriori aspetti: la velocità relativa, l'interazione con il veicolo davanti a quello davanti, l'interazione con quello dietro, la presenza di due tipi di veicoli, come ad esempio auto e camion.

2.2 Altri modelli

I modelli di velocità ottimale descritti nella sezione precedente sono modelli microscopici che descrivono il traffico per mezzo di un sistema dinamico. Vi

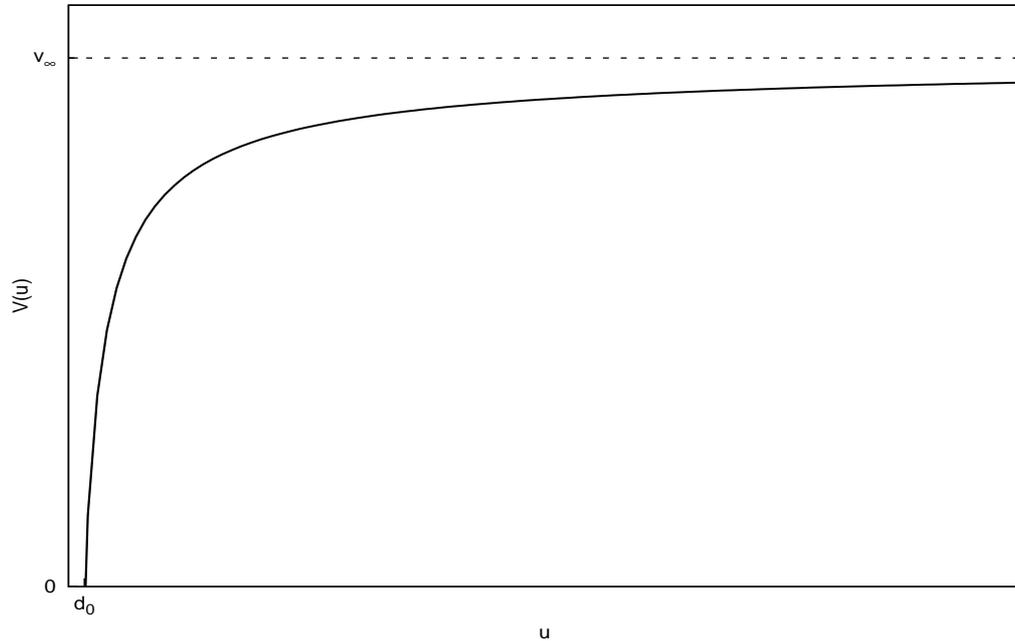


Figura 3: Grafico della funzione velocità ottimale (5) per $a = 0.75$ e $m = 1$

sono però altri modelli, sia di tipo microscopico che macroscopico. Poiché essi non verranno utilizzati in questa tesi ci limiteremo a brevissimi cenni, rinviando per esempio ad [1] per dei dettagli.

In altri tipi di modelli microscopici, proposti originariamente da Prigogine nel 1971, i veicoli vengono trattati come un gas di particelle interagenti e il traffico è descritto dall'equazione di Boltzmann.

Altri modelli microscopici per la descrizione del traffico usano automi cellulari.

Le teorie macroscopiche trattano invece il traffico come un fluido comprimibile unidimensionale. Lo stato del traffico nella posizione x al tempo t è descritto in termini di densità $\rho(x, t)$ e velocità media $v(x, t)$.

3 Studio della stabilità di una soluzione di un semplice modello di velocità ottimale

In questo capitolo applicheremo il modello differenziale del secondo ordine (2) allo studio di due modelli: nel primo le auto si muovono lungo una circonferenza, nel secondo invece lungo una retta [3], [4].

3.1 Auto che si muovono lungo una circonferenza

Il problema fisico che vogliamo studiare è quello di $N + 1$ auto che si muovono lungo una circonvallazione, modellizzata come una circonferenza. Supponiamo di avere una sola corsia. In assenza di immissioni di altre auto si possono avere situazioni di traffico stazionario in cui tutte le auto viaggiano a velocità costante mantenendo distanza costante. L'immissione di altre auto introduce una perturbazione nel sistema poiché delle auto dovranno frenare per lasciar entrare quelle nuove. Si osserva che a basse densità di veicoli questa perturbazione viene riassorbita rapidamente, mentre ad alte densità le velocità delle auto inizieranno ad oscillare attorno al valore costante iniziale fino ad arrivare ad una situazione di stop-and-go: le auto in alcuni tratti vanno alla velocità massima e poi frenano improvvisamente e percorrono il tratto seguente a velocità quasi nulla per poi riaccelerare nuovamente e così via. I modelli di velocità ottimale descrivono bene questo fenomeno.

Consideriamo quindi un insieme di $N + 1$ auto A_0, A_1, \dots, A_N che si muovono lungo la circonferenza C di lunghezza L . Poiché si muovono su una sola corsia, le auto non possono superarsi. Sia $s_j \in C$ la posizione di A_j e per comodità definiamo inoltre $s_{N+1} = s_0$. Se imponiamo il vincolo che le auto debbano mantenere una distanza di sicurezza d_0 , cioè

$$d(s_j, s_{j-1}) > d_0 \quad j = 1, \dots, N + 1$$

ove d è la distanza sul cerchio, allora lo spazio delle configurazioni del sistema è

$$Q = \{(s_0, \dots, s_N) \in C^{N+1} : d(s_{j-1}, s_j) > d_0 \quad \forall j = 1, \dots, N + 1\}.$$

La dinamica del sistema è descritta dal sistema del secondo ordine

$$\ddot{s}_j = -b(\dot{s}_j - V(s_{j-1} - s_j)) \quad j = 1, \dots, N + 1$$

Prendiamo come velocità ottimale la funzione (4), che è definita per $\Delta s > d_0$ (data la definizione dello spazio delle configurazioni non c'è bisogno di definirla per $\Delta s \leq d_0$).

Facendo il cambio di variabili

$$s_j \rightarrow s_j/d_0, \quad t \rightarrow tv_\infty/d_0,$$

e dunque $v_j \rightarrow v_j/v_\infty$, il sistema prende la forma adimensionale:

$$\ddot{s}_j = -\beta(\dot{s}_j - v_{opt}(s_{j-1} - s_j)) \quad j = 1, \dots, N+1 \quad (5)$$

dove $\beta = bd_0/v_\infty$ e

$$v_{opt}(\Delta s) = \left(1 - \left(\frac{1}{\Delta s}\right)^a\right)^m.$$

È immediato notare che il sistema (5) ha una famiglia di soluzioni stazionarie nelle quali tutte le auto viaggiano a velocità costante \bar{v} e sono equispaziate a distanza $\bar{d} = \frac{L}{N+1}$, con \bar{d} e \bar{v} univocamente fissati da $\bar{v} = v_{opt}(\bar{d})$. Tali soluzioni differiscono fra loro per una traslazione di tutte le auto (il sistema ha un'ovvia invarianza per traslazioni) e sono date da

$$s_j(t) = j\bar{d} + \bar{v}t + const \pmod{L} \quad j = 1, \dots, N+1. \quad (6)$$

L'invarianza per traslazioni permette la riduzione ad un sistema con un grado di libertà in meno e le soluzioni stazionarie del sistema originale corrispondono ad una soluzione di equilibrio del sistema ridotto.

In pratica, la riduzione si effettua prendendo come coordinate s_0, u_1, \dots, u_N con

$$u_j = s_j - s_{j+1} \pmod{L} \quad j = 1, \dots, N$$

ed ignorando la coordinata s_0 che non compare nelle equazioni del moto per le altre coordinate, che sono

$$\begin{aligned} \ddot{u}_1 &= -\beta \left(\dot{u}_1 - v_{opt}\left(L - \sum_{i=1}^N u_i\right) + v_{opt}(u_2) \right) \\ \ddot{u}_j &= -\beta(\dot{u}_j - v_{opt}(u_{j-1}) + v_{opt}(u_j)) \quad j = 2, \dots, N \end{aligned} \quad (7)$$

e sono riguardabili come definite nello spazio delle configurazioni

$$S = \{u \in C^N : u_j > d_0 \quad j = 1, \dots, N\}.$$

Le soluzioni stazionarie (6) corrispondono all'equilibrio

$$\bar{u} = (\bar{u}_1, \dots, \bar{u}_N, 0, \dots, 0), \quad \bar{u}_j = \frac{L}{d_0(N+1)} \quad \forall j$$

di questo sistema. Vogliamo ora studiare il problema della stabilità lineare di questo equilibrio.

Proposizione. Sia $K = \frac{dv_{opt}}{du}(\bar{u})$. Se $\beta > 2K$, allora l'equilibrio $(\bar{u}, \dots, \bar{u}, 0, \dots, 0)$ è asintoticamente stabile. Se $\beta < 2K$, allora è instabile.

Dimostrazione. Scriviamo il sistema (7) come sistema del primo ordine e linearizziamolo attorno all'equilibrio. Otteniamo:

$$\begin{aligned} \dot{u}_1 &= v_1 \\ \dot{u}_2 &= v_2 \\ &\vdots \\ \dot{u}_N &= v_N \\ \dot{v}_1 &= -\beta \left[v_1 - \sum_{j=2}^N K(u_j - \bar{u}) \right] \\ \dot{v}_2 &= -\beta [v_2 + K(u_2 - \bar{u}) - K(u_1 - \bar{u})] \\ &\vdots \\ \dot{v}_N &= -\beta [v_N + K(u_N - \bar{u}) - K(u_{N-1} - \bar{u})] \end{aligned}$$

Ponendo $x_j = u_j - \bar{u}$, otteniamo un sistema omogeneo che possiamo scrivere con una notazione compatta come

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{con} \quad A = \begin{pmatrix} 0 & \mathbb{I} \\ B & -\beta\mathbb{I} \end{pmatrix}$$

dove ogni termine rappresenta un blocco $N \times N$ e

$$B = \beta K \begin{pmatrix} 0 & 1 & \dots & \dots & 1 \\ 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \ddots & 1 & -1 \end{pmatrix}$$

Dobbiamo ora determinare gli autovalori di A. A tal fine usiamo la definizione: $\lambda \in \mathbb{C}$ è autovalore di A se esiste un vettore

$$\begin{pmatrix} w \\ z \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

tale che

$$A \begin{pmatrix} w \\ z \end{pmatrix} = \lambda \begin{pmatrix} w \\ z \end{pmatrix},$$

ovvero

$$\begin{pmatrix} z \\ Bw - \beta z \end{pmatrix} = \lambda \begin{pmatrix} w \\ z \end{pmatrix}$$

da cui otteniamo

$$Bw = (\beta\lambda + \lambda^2)w$$

Dunque, se b è autovalore della matrice B , allora ogni λ tale che $b = \beta\lambda + \lambda^2$ è autovalore della matrice A , e viceversa. Dunque se indichiamo con b_1, \dots, b_N gli autovalori di B , allora gli autovalori di A sono

$$\lambda_{\pm}^k = -\frac{\beta}{2} \left(1 \pm \sqrt{1 + \frac{4b_k}{\beta^2}} \right) \quad k = 1, \dots, N$$

con la convenzione che la radice di un numero complesso sia quella con parte reale positiva.

Per un noto teorema di Lyapunov (metodo spettrale), si ha che un equilibrio è asintoticamente stabile se la matrice A ha tutti gli autovalori con parte reale negativa, instabile se ha almeno un autovalore con parte reale positiva.

Nel nostro caso $\Re\lambda_+^k < 0 \forall k$, mentre $\Re\lambda_-^k < 0$ se $\Re\sqrt{1 + \frac{4K}{\beta}b_k} < 1$. Dunque la condizione di stabilità è

$$\Re\sqrt{1 + \frac{4K}{\beta}b_k} < 1 \quad k = 1, \dots, N. \quad (8)$$

Cerchiamo dunque gli autovalori della matrice B . Se per semplicità di notazione scriviamo $B = \beta K \hat{B}$, allora gli autovalori di B sono $b_k = \beta K \hat{b}_k$ con \hat{b}_k autovalore di \hat{B} . Per determinare questi autovalori procediamo come è d'uso per ben noti sistemi di masse e molle: cerchiamo un sistema di N soluzioni indipendenti y_1, \dots, y_N del sistema lineare del primo ordine $\dot{y} = \hat{B}y$ nella forma

$$y_k(t) = c_k e^{\hat{b}_k t} \quad k = 1, \dots, N$$

con

$$c_k = \begin{pmatrix} e^{i\frac{2\pi}{N}k} \\ e^{2i\frac{2\pi}{N}k} \\ \vdots \\ e^{i2\pi k} \end{pmatrix} \in \mathbb{C}^N \quad k = 1, \dots, N.$$

Sostituendo queste soluzioni di prova nel sistema $\dot{y} = \hat{B}y$ si trova

$$\hat{b}_k = \exp(i\frac{2\pi k}{N} - 1) \quad k = 1, \dots, N.$$

Questo è immediato per $k = 2, \dots, N$; per $k = 1$ si arriva a questa espressione di \hat{b}_1 usando l'identità trigonometrica

$$e^{i\alpha} + e^{2i\alpha} + \dots + e^{Ni\alpha} = 0 \quad (\alpha = \frac{2\pi}{N}),$$

che vale per ogni $N > 1$. Possiamo dunque riscrivere la condizione di stabilità (8) come

$$\Re \sqrt{1 + \frac{4K}{\beta} \left(e^{-i\frac{2\pi k}{N}} - 1 \right)} < 1 \quad \forall k = 1, \dots, N. \quad (9)$$

Osservo ora che, se $z \in \mathbb{C} \setminus \{0\}$ e \sqrt{z} denota la radice quadrata di z con parte reale positiva, allora si ha

$$\Re(\sqrt{z}) < 1 \iff |z| < 2 - \Re(z).$$

Infatti scrivendo $\sqrt{z} = |\sqrt{z}|e^{i\phi}$ si ha $(\Re(\sqrt{z}))^2 = |z| \cos^2(\phi) = |z| \frac{1+\cos(\phi)}{2}$ e dunque $\Re(\sqrt{z}) < 1 \iff (\Re(\sqrt{z}))^2 < 1 \iff |z| < 2 - \Re(z)$.

Allora se per k fissato poniamo $A = \frac{4K}{\beta}$ e $\theta = -\frac{2\pi k}{N}$, la condizione di stabilità (9) si scrive $\Re \sqrt{1 + Ae^{i\theta} - A} < 1$ ed equivale a

$$\sqrt{(1 - A + A \cos \theta)^2 + A^2 \sin^2 \theta} < 1 + A - A \cos \theta.$$

Dopo un po' di passaggi algebrici si ottiene che questa condizione equivale a $A(1 - \cos \theta)(4 - A(\cos \theta + 1)) > 0$, cioè $\cos \theta < \frac{4}{A} - 1$. Quindi la condizione di stabilità è

$$\cos \frac{2\pi k}{N} < \frac{\beta}{K} - 1, \quad k = 1, \dots, N.$$

Poiché il coseno ha il massimo in $0 \equiv 2\pi$, queste condizioni sono tutte soddisfatte se e solo se

$$1 < \frac{\beta}{K} - 1 \quad \text{cioè} \quad \beta > 2K.$$

Se invece $\beta < 2K$, allora la matrice A ha N autovalori con parte reale positiva e quindi l'equilibrio è instabile. \square

3.2 Auto che si muovono lungo una retta

Consideriamo $N + 1$ auto che si muovono lungo una linea retta. Sia s_j la posizione dell'auto j -esima, $j = 0, \dots, N$. Il moto della prima auto, che è chiamata "leading car" e ha indice 0, è assegnato come funzione nota $s_0(t)$ del tempo. Dunque lo spazio delle configurazioni è costituito dalle posizioni s_1, \dots, s_N con il vincolo $s_{j-1} - s_j > d_0 \forall j$, e le equazioni del moto sono un sistema di N equazioni differenziali non autonome:

$$\begin{aligned}\ddot{s}_1 &= -\beta (\dot{s}_1 - v_{opt}(s_0(t) - s_1)) \\ \ddot{s}_j &= -\beta (\dot{s}_j - v_{opt}(s_{j-1} - s_j)) \quad j = 2, \dots, N\end{aligned}$$

Assumiamo che la "leading car" si muova a velocità costante, cioè

$$s_0(t) = s_0(0) + vt.$$

Facendo il cambio di variabili dipendenti dal tempo

$$\begin{aligned}u_1 &= s_0(t) - s_1 \\ u_j &= s_{j-1} - s_j \quad j = 2, \dots, N\end{aligned}$$

otteniamo il sistema

$$\begin{aligned}\ddot{u}_1 &= -\beta (\dot{u}_1 - v + v_{opt}(u_1)) \\ \ddot{u}_j &= -\beta (\dot{u}_j + v_{opt}(u_j) - v_{opt}(u_{j-1})) \quad j = 2, \dots, N\end{aligned}$$

In questo caso abbiamo, per ogni valore della velocità della leading car, infinite soluzioni di equilibrio date da $(\bar{u}, \dots, \bar{u}, 0, \dots, 0)$, dove \bar{u} è un qualsiasi valore costante $> d_0$.

Si noti che, a differenza del caso delle auto sul cerchio, vi sono ora soluzioni stazionarie con distanza (e dunque velocità) arbitrarie. (Nel caso del cerchio distanza e velocità erano rese uniche dal fatto che la prima auto "segue" l'ultima).

Proposizione. L'equilibrio $(\bar{u}, \dots, \bar{u}, 0, \dots, 0)$ è asintoticamente stabile $\forall \beta$.

Dimostrazione. Scriviamo il sistema come sistema del primo ordine e

linearizziamolo attorno all'equilibrio. Otteniamo:

$$\begin{aligned}
 \dot{u}_1 &= v_1 \\
 \dot{u}_2 &= v_2 \\
 &\vdots \\
 \dot{u}_N &= v_N \\
 \dot{v}_1 &= -\beta [v_1 + K(u_1 - \bar{u})] \\
 \dot{v}_2 &= -\beta [v_2 + K(u_2 - \bar{u}) - K(u_1 - \bar{u})] \\
 &\vdots \\
 \dot{v}_N &= -\beta [v_N + K(u_N - \bar{u}) - K(u_{N-1} - \bar{u})]
 \end{aligned}$$

Procedendo come nella dimostrazione del caso del cerchio, otteniamo risultati analoghi, con la differenza che in questo caso la matrice B è

$$B = \beta K \begin{pmatrix} -1 & 0 & \dots & \dots & 0 \\ 1 & -1 & \ddots & & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 & 0 \\ 0 & \dots & 0 & 1 & -1 \end{pmatrix}$$

Questa matrice ha autovalore -1 con molteplicità N , e dunque gli autovalori del sistema linearizzato sono:

$$\lambda_{\pm}^j = -\frac{\beta}{2} \left(1 \pm \sqrt{1 - \frac{4K}{\beta}} \right) \quad j = 1, \dots, N$$

Se $\beta \geq 4K$ tutti gli autovalori sono reali negativi, se $\beta < 4K$ essi sono complessi con parte reale negativa. Per il primo teorema di Lyapunov l'equilibrio è quindi asintoticamente stabile $\forall \beta$. \square

4 Stop-and-go

Vogliamo ora capire come varii la dinamica al variare della densità di veicoli. Nel caso del cerchio se aumenta la densità, diminuisce il valore di \bar{u} e di conseguenza aumenta $K = \frac{dv_{opt}}{du}(\bar{u})$ e dunque il rapporto $\frac{\beta}{2K}$ diminuisce facendo sì che si passi da una soluzione stazionaria stabile ad una instabile.

4.1 Simulazioni Numeriche

Ho effettuato delle simulazioni numeriche, scrivendo un programma in C++ che utilizza per l'integrazione del sistema di equazioni differenziali un algoritmo di Runge-Kutta del IV ordine [6]. I programmi sono riportati in appendice. Per passare dal caso instabile a quello stabile per comodità in queste simulazioni invece di cambiare N (e quindi la densità di veicoli), ho cambiato β . I parametri utilizzati sono i seguenti:

- $N = 1000$
- $L = 2000$;
- $a = 0.75$ e $m = 1$;

In tutti i casi si è trovato adeguato utilizzare un passo temporale di 0.05 unità temporali (u.t.).

4.1.1 Auto che si muovono lungo una circonferenza

Per simulare le perturbazioni dovute all'entrata e uscita di auto dalla circonvallazione, ho usato una distribuzione sinusoidale delle distanze iniziali tra le auto:

$$u_j(0) = \frac{L}{N} + \epsilon \sin\left(\frac{2\pi j}{N}\right), \quad \epsilon = 0.05, \quad j = 1, \dots, N$$

I grafici della Figura 4 sono stati ottenuti in condizione di instabilità con $\beta = 0.9\beta_{critico}$, mentre i grafici di Figura 5 in condizione di stabilità con $\beta = 2\beta_{critico}$, con $\beta_{critico} = 2K = \frac{2a}{u^{a+1}} = 0.446$.

Dai grafici di Figura 4 risulta evidente che in una situazione di instabilità una perturbazione (pur piccola) del sistema provoca un'oscillazione della velocità delle auto attorno al valore della velocità ottimale di equilibrio (nel nostro caso pari a 0.405) che diventa sempre più marcata fino a portare ad una dinamica di tipo stop-and-go, che permane indefinitamente.

Nel caso stabile (Figura 5) invece la perturbazione non aumenta in modo significativo e il valore della velocità delle auto rimane attorno al valore della velocità ottimale all'equilibrio.

4.1.2 Auto che si muovono lungo una retta

In questo caso consideriamo una situazione nella quale la perturbazione è dovuta al fatto che la "leading car" dimezza improvvisamente la sua velocità rispetto a tutte le altre e dopo $25u.t.$ torna a viaggiare alla velocità che aveva prima della frenata.

In accordo col fatto che la soluzione è sempre stabile osserviamo come la perturbazione si propaghi all'indietro come un'onda, fino a scomparire.

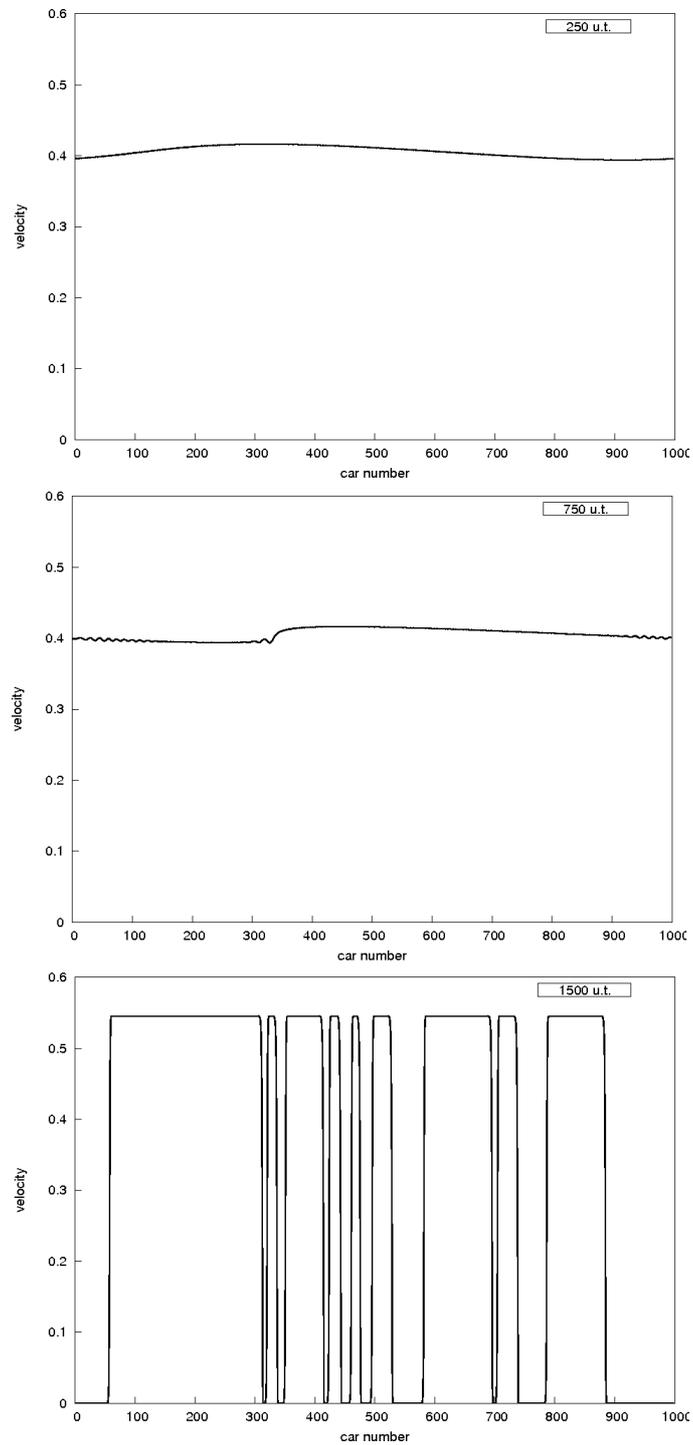


Figura 4: Caso instabile sul cerchio: distribuzione delle velocità in tre diversi istanti

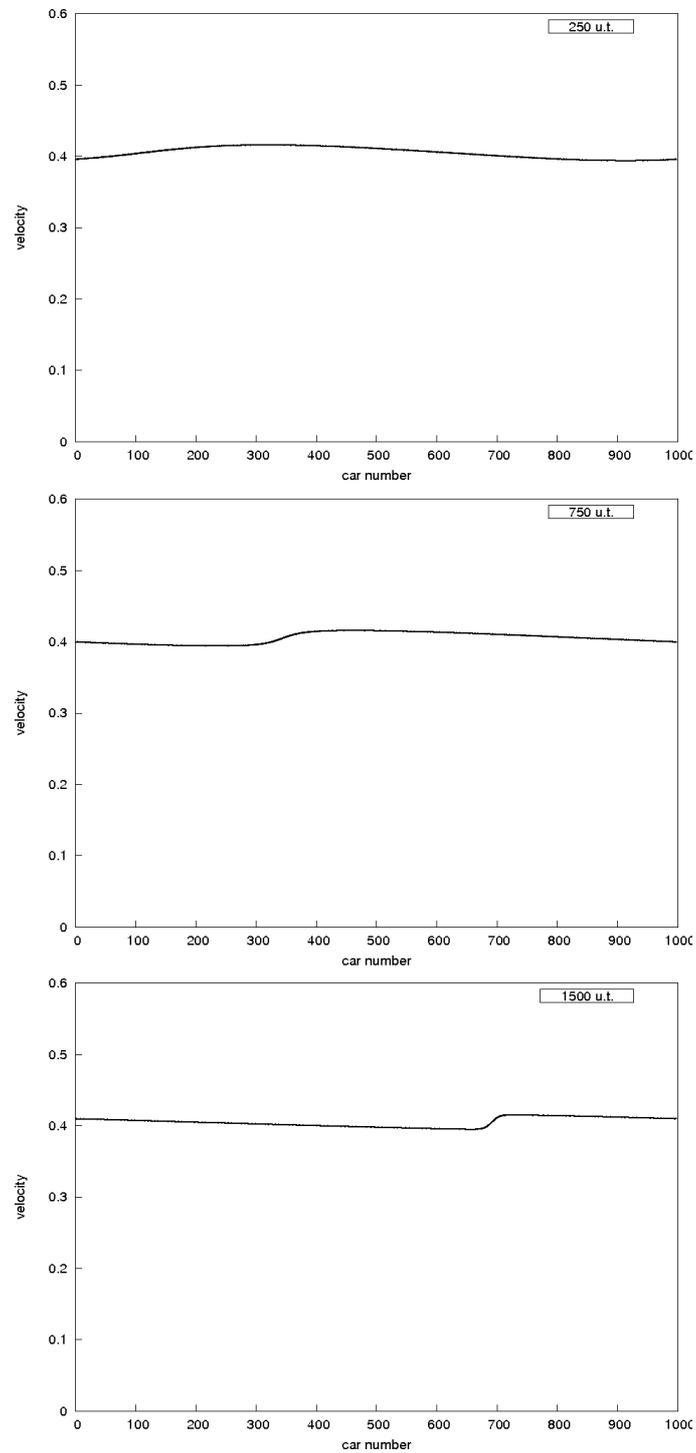


Figura 5: Caso stabile sul cerchio: distribuzione delle velocità in tre diversi istanti

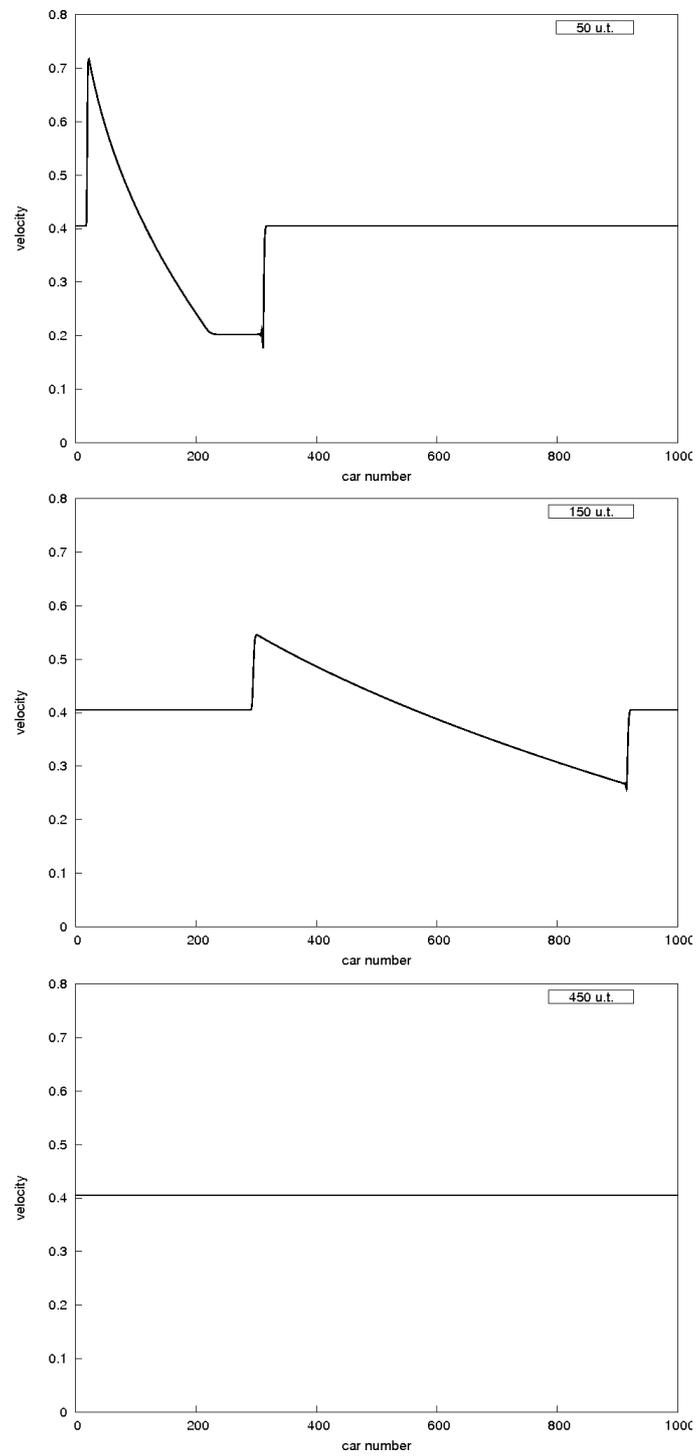


Figura 6: Sulla retta: distribuzione delle velocità in tre diversi istanti

5 Appendice

5.1 Codice del programma che simula il moto di n auto lungo una circonferenza

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <iomanip>

using namespace std;

const int N = 2000; //Numero di equazioni del sistema del primo ordine
const double L = 2000.; //Lunghezza della circonferenza
const int n = 1000; //Numero di auto
const double DE = L/n; //Densita' di auto
const double a = 0.75;
const double b = 2*2.*a*pow(1./DE,a+1.); //beta = 0.9 x beta_critico
const double Vopt = 1.-pow(1./DE,a); //velocita' ottimale all'equilibrio

void runge4(double t, double y[], double step);
double f(double t, double y[], int i);

int main()
{

    cout << "beta = " << b << endl;
    cout << "Vopt = " << Vopt << endl;

    double t, y[N];
    int i;
    double dist = 0.05; //time step

    //condizioni iniziali
    y[0]=L-DE;
    for(i=1;i<n;i++){y[i]=y[i-1]-DE - 0.05*sin(2.*M_PI*i/n);}
    for(i=n;i<N;i++){y[i]=Vopt;}

    ofstream outfile;
    outfile.open("traffic");
    for(t=0;t<30000;t+=dist)
    {
        runge4(t,y,dist);
        for(i=0;i<n;i++){if(y[i]>=L) y[i] = y[i]-L;}
    }
    for(i=0;i<n;i++){outfile << y[i] << "\t" << y[i+n] << endl;}
    outfile.close();
}
```

```

return 0;

}

// Routine di Runge-Kutta
void runge4(double t, double y[], double step)
{
    double f(double t, double y[], int i);
    double h=step/2.0,
           t1[N], t2[N], t3[N],
           k1[N], k2[N], k3[N],k4[N];
    int i;

    for (i=0; i<N; i++) t1[i] = y[i]+0.5*(k1[i]=step*f(t, y, i));
    for (i=0; i<N; i++) t2[i] = y[i]+0.5*(k2[i]=step*f(t+h, t1, i));
    for (i=0; i<N; i++) t3[i] = y[i]+      (k3[i]=step*f(t+h, t2, i));
    for (i=0; i<N; i++) k4[i] =                step*f(t + step, t3, i);

    for (i=0; i<N; i++) y[i] += (k1[i]+2*k2[i]+2*k3[i]+k4[i])/6.0;
}

//Definizione del sistema di equazioni
double f(double t, double y[], int i)
{
    if (i < n) return (y[i+n]);
    if (i == n)
        { if(y[0]<y[n-1]) {if (y[n-1]-y[0] <= 1. ) return (-b*y[n]);
                        else return (-b*(y[n]-(1.-pow(1./(y[n-1]-y[0]),a))));}
          else { if (y[n-1]+L-y[0] <= 1. ) return (-b*y[n]);
                else return (-b*(y[n]-(1.-pow(1./(y[n-1]+L-y[0]),a))));}
        }
    if (i > n )
        { if (y[i-n]>y[i-n-1] ) { if (y[i-n-1]+L-y[i-n] <= 1. ) return (-b*y[i]);
                                else return (-b*(y[i]-(1.-pow(1./(y[i-n-1]+L-y[i-n]),a))));}
          else { if (y[i-n-1]-y[i-n] <= 1. ) return (-b*y[i]);
                else return (-b*(y[i]-(1.-pow(1./(y[i-n-1]-y[i-n]),a))));}
        }
}
}

```

5.2 Codice del programma che simula il moto di n auto lungo una linea retta

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <iomanip>

using namespace std;

const int N = 2000; //Numero di equazioni del sistema del primo ordine
const double L = 2000.; //Lunghezza segmento lungo cui sono distribuite le auto a t=0
const int n = 1000; //Numero di auto
const double DE = L/n; //Densit  iniziale
const double a = 0.75;
const double b = 3.6*a*pow(1./DE,a+1.);
const double Vopt = 1.-pow(1./DE,a); //Velocit  ottimale alla densit  iniziale

void runge4(double t, double y[], double step);
void runge4bis(double t, double y[], double step);
double f(double t, double y[], int i);
double g(double t, double y[], int i);
double x(double t);
double z(double t);

int main()
{

    cout << "beta = " << b << endl;
    cout << "Vopt = " << Vopt << endl;

    double t, y[N];
    int i;
    double dist = 0.05;

    //condizioni iniziali
    y[0]=L-DE;
    for(i=1;i<n;i++){y[i]=y[i-1]-DE;}
    for(i=n;i<N;i++){y[i]=Vopt;}

    ofstream outfile;
    outfile.open("traffic_R");
    for(t=0;t<=500.;t+=dist)
    {
        runge4(t,y,dist);
    }
    for(t=500.05;t<=9000;t+=dist)
    {
        runge4bis(t,y,dist);
    }
}
```

```

}
outfile << z(t) << "\t" << Vopt << endl;
for(i=0;i<n;i++){outfile << y[i] << "\t" << y[i+n] << endl;}
outfile.close();

return 0;

}

void runge4(double t, double y[], double step)
{
    double f(double t, double y[], int i);
    double h=step/2.0,
           t1[N], t2[N], t3[N],
           k1[N], k2[N], k3[N],k4[N];
    int i;

    for (i=0; i<N; i++) t1[i] = y[i]+0.5*(k1[i]=step*f(t, y, i));
    for (i=0; i<N; i++) t2[i] = y[i]+0.5*(k2[i]=step*f(t+h, t1, i));
    for (i=0; i<N; i++) t3[i] = y[i]+      (k3[i]=step*f(t+h, t2, i));
    for (i=0; i<N; i++) k4[i] =
                               step*f(t + step, t3, i);

    for (i=0; i<N; i++) y[i] += (k1[i]+2*k2[i]+2*k3[i]+k4[i])/6.0;
}

double f(double t, double y[], int i)
{
    if (i < n) return (y[i+n]);
    if (i == n) {if (x(t)-y[0] <= 1. ) return (-b*y[n]);
                else return (-b*(y[n]-(1.-pow(1./(x(t)-y[0]),a)))));}
    if (i > n ) { if (y[i-n-1]-y[i-n] <= 1. ) return (-b*y[i]);
                else return (-b*(y[i]-(1.-pow(1./(y[i-n-1]-y[i-n]),a)))));}
}

double x(double t)
{
    return 0.5*Vopt*t + L;
}

void runge4bis(double t, double y[], double step)
{
    double g(double t, double y[], int i);
    double h=step/2.0,
           t1[N], t2[N], t3[N],
           k1[N], k2[N], k3[N],k4[N];
    int i;

    for (i=0; i<N; i++) t1[i] = y[i]+0.5*(k1[i]=step*g(t, y, i));
    for (i=0; i<N; i++) t2[i] = y[i]+0.5*(k2[i]=step*g(t+h, t1, i));
}

```

```

    for (i=0; i<N; i++) t3[i] = y[i]+    (k3[i]=step*g(t+h, t2, i));
    for (i=0; i<N; i++) k4[i] =          step*g(t + step, t3, i);

    for (i=0; i<N; i++) y[i] += (k1[i]+2*k2[i]+2*k3[i]+k4[i])/6.0;
}

double g(double t, double y[], int i)
{
    if (i < n) return (y[i+n]);
    if (i == n) {if (z(t)-y[0] <= 1. ) return (-b*y[n]);
                 else return (-b*(y[n]-(1.-pow(1./(z(t)-y[0]),a)))));}
    if (i > n ) { if (y[i-n-1]-y[i-n] <= 1. ) return (-b*y[i]);
                 else return (-b*(y[i]-(1.-pow(1./(y[i-n-1]-y[i-n]),a)))));}
}

double z(double t)
{
    return Vopt*t + L;
}

```


Riferimenti bibliografici

- [1] T. Nagatani: *The physics of traffic jams*, Rep. Prog. Phys. 65 1331-1386 (2002)
- [2] G. F. Newell: *Nonlinear effects in the dynamics of car following*, Oper. Res. 9 209 (1961)
- [3] M. Bando, K. Hasebe, A. Nakayama, A. Shibata and Y. Sugiyama: *Dynamical model of traffic congestion and numerical simulation*, Phys. Rev. E 51 1035 (1995)
- [4] A. Bazzani: *Remark on a simple optimal velocity model*, Comunicazione privata
- [5] F. Fassò: *Dispense per il corso di Istituzioni di Fisica Matematica*, Università di Padova, Note non pubblicate
- [6] A. Crisanti: *yaC Primer, Parte II: Applicazioni*, Ed. Aracne (2006)