

Relazione 4: Soluzione numerica Problemi di Cauchy

Antonio Michele Miti

12 gennaio 2010

Sommario

In questo articolo si intende mostrare come implementare alcuni algoritmi numerici per il calcolo della soluzione dei problemi di Cauchy di primo ordine stimandone graficamente l'ordine di approssimazione.

Inoltre viene descritta la possibile estensione di tali metodi ad equazioni differenziali ordinarie ad ordini superiori al primo al fine di applicarlo allo studio della dinamica del Pendolo Caotico.

L'interesse di questo sistema fisico sta nella sua forte dipendenza dai parametri iniziali, tale da mostrare un comportamento pseudo-caotico attorno a particolari valori critici.

1 Introduzione

Dato il problema di Cauchy

$$\begin{cases} \dot{y}(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \quad (1)$$

di soluzione

$$y(t) = y_0 + \int_{t_0}^t f(x, y(x)) dx \quad (2)$$

si è interessati a trovare una soluzione numerica del problema.

In altre parole si cerca di ottenere con una certa approssimazione un samplaggio, valutazione della funzione in un numero discreto $\{t_i\}_{i \in \mathbb{N}}$ di punti del dominio, della funzione soluzione.

Analiticamente vale che

$$y(t_{i+1}) = y_0 + \int_{t_i}^{t_{i+1}} f(x, y(x)) dx = y_0 + h_i \int_0^1 f(t_0 + \theta h_i, y(t_0 + \theta h_i)) d\theta h \quad (3)$$

a seguito del cambio di variabili

$$\begin{cases} x = t_i + \theta(t_{i+1} - t_i) \\ t_{i+1} - t_i = h_i \end{cases}$$

I metodi di approssimazione entrano in gioco nel calcolo numerico del integrale precedente.

Scegliendo un ordine di quadratura ν si ottiene:

$$y(t_{i+1}) \simeq y_{i+1} = y_i + h_i \sum_{j=0}^{\nu} \omega_j f(t_0 + \theta_j h_i, y(t_0 + \theta_j h_i)) = y_i + \sum_{j=0}^{\nu} \omega_j K_{ij} \quad (4)$$

Dove $K_{ij} = h_i f(t_0 + \theta_j h_i, \tilde{y}(t_0 + \theta_j h_i))$ è una matrice dipendente complessivamente da come viene effettuato il campionamento del set di punti del sampling (coefficiente h_i), da come sono presi i ν punti in ciascuno step di integrazione (coefficiente θ_j dove ω_j è il relativo peso) e da come viene approssimata la funzione $y(t_0 + \theta_j h_i)$ il cui valore non è noto a priori. In sostanza discrimina tra i diversi metodi di integrazione di equazioni differenziali che hanno lo stesso ordine di quadratura.

Tutti questi metodi basati sul calcolo della soluzione attraverso il calcolo di un integrale dipendono in sostanza da 4 parametri arbitrari:

1. La scelta dei punti $\{t_i\}$ in cui campionare la soluzione. Ad esempio:
 - equidistante: $t_i = t_0 + i\tau$, dove τ è il passo del campionamento
 - adattivo : il campionamento viene infittito in alcune zone problematiche del dominio.
2. ν : ordine della quadratura, numero di punti che vengono valutati nell'intervallo.
3. Da come la funzione $f(x, y)$ viene approssimata nel calcolo dell'integrale

$$f(t_0 + \theta_j h_i, y(t_0 + \theta_j h_i)) \simeq f(t_0 + \theta_j h_i, \tilde{y}(t_0 + \theta_j h_i)) = \tilde{f}(\theta_j, h_i, i) = \tilde{f}_{ij}$$

4. dalla coppia $\{\theta_j, \omega_j\}$, dove:
 - θ_j indica come vengono presi i ν punti della quadratura nel intervallo $I = [t_i, t_{i+1}]$.
 - ω_j è il peso con cui i valori nei ν punti vengono sommati.

1.1 Metodi Runge-Kutta

La logica di questi metodi consiste di cercare la soluzione del problema di cauchy [1] come uno sviluppo in serie di Taylor nella forma:

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2!}y''_n + \frac{h^3}{3!}y'''_n \dots \quad (5)$$

approssimando y_{n+1} come combinazione

$$y_{n+1} = y_n + h[\omega_0 f(x_n, y_n) + \omega_1 f(x_n + \theta_1 h, y_n + \alpha_1 h) + \dots + \omega_p f(x_n + \theta_p h, y_n + \alpha_p h)] \quad (6)$$

dove i parametri ω , θ e α vengono scelti in modo che lo sviluppo di Taylor della funzione [6] coincida fino ad un certo ordine, il più alto possibile, con l'espansione [5].

E' evidente che tali metodi rappresentino solo una parte ristretta dei metodi più generali descritti prima, in questo caso infatti l'approssimazione della funzione y , nel calcolo di $f(x, y)$, è esplicitata.

2 Metodologia

Senza mostrare la costruzione generale degli algoritmi di Runge-Kutta è possibile presentare 3 algoritmi: eulero, costruttore e Runge-Kutta ordine IV.

Anche se apparentemente diversi, questi metodi sono tutti imparentati con l'idea di approssimare la funzione con il suo incremento finito. L'ordine del metodo corrisponde in qualche modo con l'ordine a cui lo sviluppo troncato della soluzione teorica approssima lo sviluppo troncato della soluzione numerica, per questa proprietà fanno parte di diritto dei cosiddetti metodi Runge-Kutta.

Fatto questo si può scrivere un programma *C++* che permetta di far emergere fino a che ordine dello sviluppo gli algoritmi presi in considerazione approssimano la soluzione del problema [1], con lo scopo di determinare quale dei tre algoritmi ha la maggiore precisione per poi di utilizzarlo per le considerazioni successive.

In realtà non si darà un vera e propria stima della precisione, ma più precisamente verrà calcolato l'ordine dello sviluppo di Taylor a cui la soluzione numerica e quella teorica coincidono maggiormente.

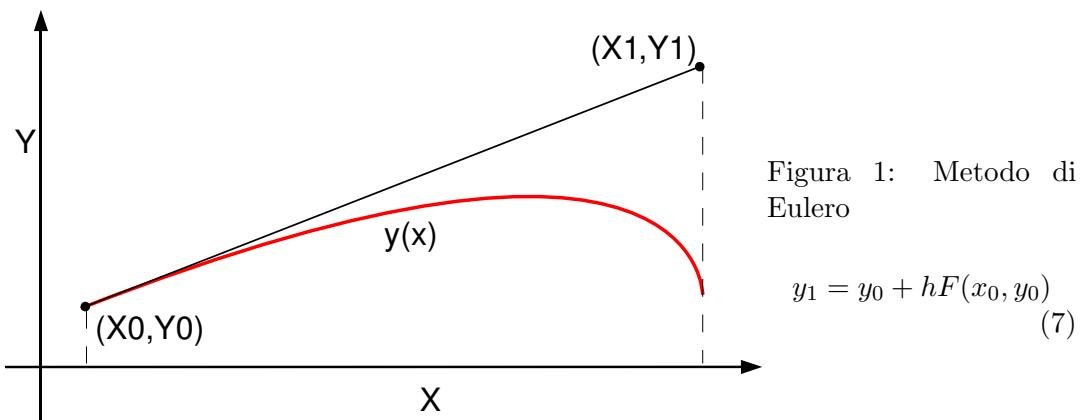
Non c'è un'implicazione diretta tra ordine del metodo e dispersione tra il risultato ottenuto nei singoli punti rispetto alla soluzione attesa, questa infatti risente principalmente dalla regolarità della singola funzione presa in considerazione.

Per avere migliore efficienza computazionale sarebbe necessario affinare l'algoritmo sulla classe di equazioni differenziali che si vuole integrale, non potendo fare questo si ricorre ai metodi Runge-Kutta, che per quanto non siano sempre i più efficienti, sono ottimi per approcciare brutalmente la maggior parte dei problemi.

Per applicare tutto questo al caso del pendolo caotico, è necessario estendere l'algoritmo scelto per la soluzione di O.D.E. di ordine superiore al primo.

2.1 Metodo di Eulero

E' il più semplice metodo di integrazione di equazioni differenziali. Conoscendo un punto (x_0, y_0) della soluzione del problema (1) tale metodo consiste nel approssimare il valore della soluzione in un punto $x_1 = x_0 + h$ con la formula dell'incremento finito della funzione al primo ordine.



Dalla definizione è evidente la sua relazione con gli algoritmi di Runge-Kutta, non è altro che il metodo di ordine 1, dove i parametri α e θ non entrano ancora

in gioco (per questo il metodo di primo ordine è unico) e il parametro di peso è unitario visto che per ogni intervallo di campionamento il punto utilizzato per calcolare l'approssimazione dello step successivo è solo uno.

2.2 Metodo del Correttore

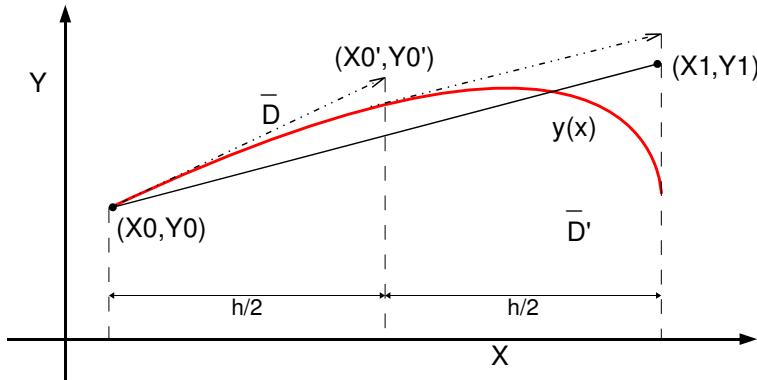
Consiste sostanzialmente in un affinamento del metodo precedente.

Per stimare la soluzione nel punto $x_0 + h$ non viene utilizzato il valore della derivata nel punto x_0 , che viene fornita attraverso la funzione $f(x, y)$ dell'equazione differenziale, ma viene utilizzata nella formula del incremento finito una stima del valore della derivata nel punto medio dell'intervallo.

$$y_1 = y_0 + hF(x'_0, y'_0) \quad (8)$$

dove $x'_0 = x_0 + \frac{h}{2}$ e $y'_0 = y_0 + \frac{h}{2}F(x_0, y_0)$.

Figura 2: Metodo del Correttore



La ragione di questo nome è nel fatto che il valore della funzione dopo l'incremento h non viene stimata con la derivata nel punto iniziale ma viene preso una correzione di tale valore, la derivata nel punto medio.

Anche questo metodo fa parte della classe dei metodi R-K, l'ordine è 2 per stimare il punto destro dell'intervallo vengono presi 2 punti all'interno, il primo è l'estremo sinistro con peso nullo e il secondo è il punto medio con peso unitario. I parametri nella formula di R-K generico (6) sono ($\omega_0 = 0$) e ($\omega_1 = 1, \theta_1 = \frac{1}{2}, \alpha_1 = f(x_0, y_0)$)

2.3 Runge Kutta di quarto ordine

Esistono diversi metodi di integrazione che stimano la soluzione del problema (1) in un punto valutando la funzione f in 4 punti dell'intervallo precedente, l'algoritmo utilizzato usualmente è il seguente:

$$\begin{aligned} K_0 &= hf(x_0, y_0) \\ K_1 &= hf(x_0 + h/2, y_0 + K_0/2) \\ K_2 &= hf(x_0 + h/2, y_0 + K_1/2) \\ K_3 &= hf(x_0 + h, y_0 + K_2) \end{aligned}$$

$$y_1 = y_0 + (K_0 + 2K_1 + 2K_2 + K_3)/6 \quad (9)$$

Dalla definizione è evidente la sua parentela con l'idea fondamentale dell'incremento finito.

Ad ogni step dell'algoritmo la funzione $f(x, y)$, in altre parole la derivata prima, viene stimata in quattro punti, una volta all'inizio, due volte nel mezzo e una volta nell'estremo destro dell'intervallo dell'ascissa.

Alla fine l'incremento finito della soluzione dopo lo step di ampiezza h viene approssimato mediando in modo pesato le 4 derivate prime calcolate.

Figura 3: Metodo Runge Kutta : progressione dei punti valutati al IV ordine
includegraphics[width=8cm,height= 4cm]picture/runge

Come tutti gli altri metodi R-K è discreto, iterativo e a singolo passo ma ha la proprietà aggiuntiva di avere una struttura particolarmente semplice: i termini K_n sono calcolati in successione, l' n -simo coefficiente dipende unicamente dall'(n-1)-simo. Questi algoritmi possono generalizzarsi all'ordine n nella forma:

$$\begin{aligned} K_0 &= hf(x_0, y_0) \\ K_1 &= hf(x_0 + \theta_1 h, y_0 + \theta_1 K_0) \\ &\vdots \\ K_n &= hf(x_0 + \theta_n h, y_0 + \theta_n K_{n-1}) \\ y_1 &= y_0 + \sum_{j=0}^{\nu} \omega_j K_j \end{aligned} \quad (10)$$

Convenzionalmente ci si riferisce a questo algoritmo come Runge-Kutta di quarto ordine.

2.4 Ordine di Correttezza degli algoritmi

Come detto precedentemente, per ordine di un algoritmo di integrazione di O.D.E. si intende fino a che ordine dello sviluppo di taylor la soluzione approssimata ricalca la soluzione attesa. Un possibile modo per stimare questo fattore è quello di integrare numericamente problemi di cauchy del tipo:

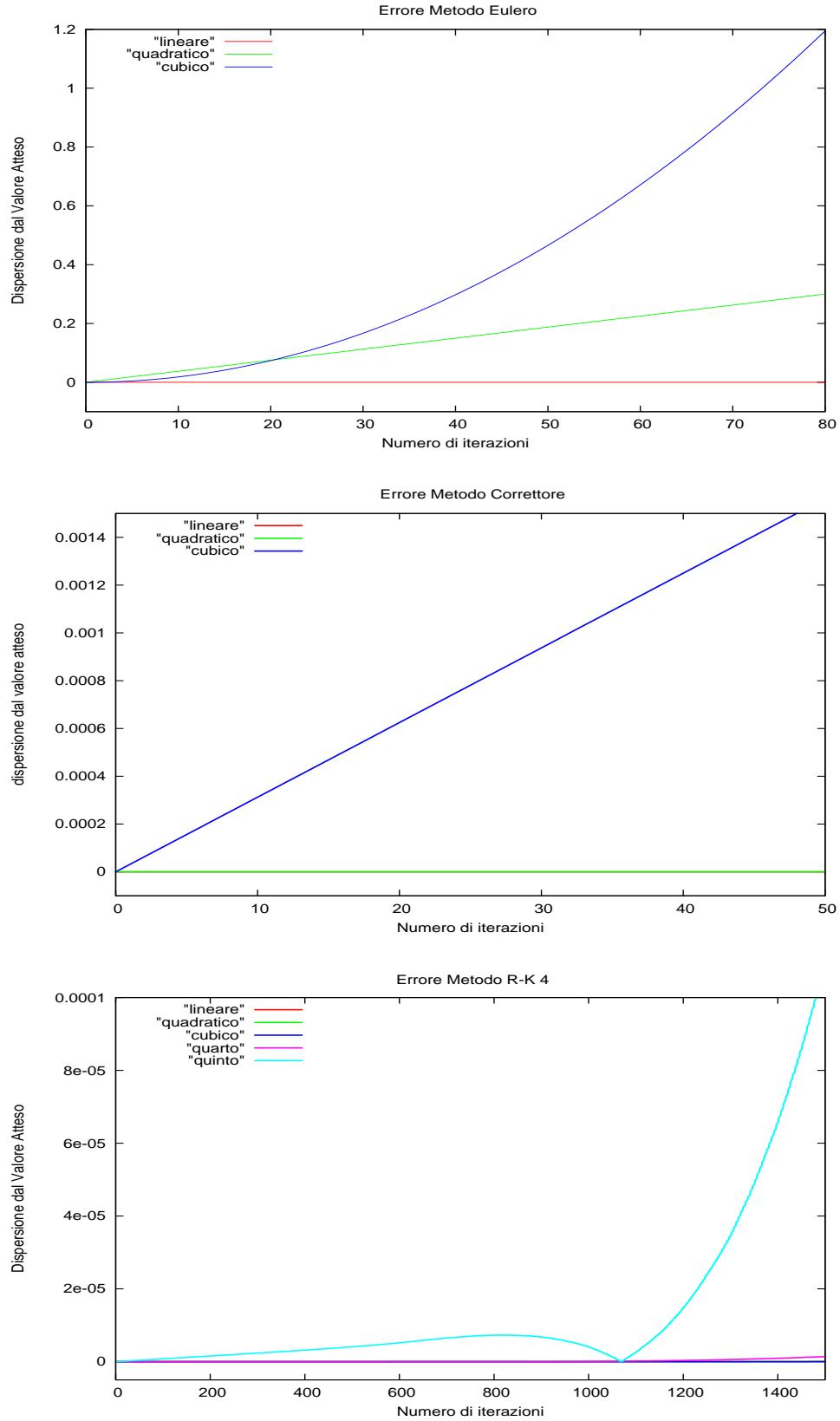
$$\begin{cases} \dot{y}(t) = \alpha x^{n-1} \\ y(t_0) = y_0 \end{cases} \quad (11)$$

la soluzione di questa equazione è una potenza ennesima di x, $y = \frac{\alpha x^n}{n} + y_0$.

A questo punto stampando per ogni metodo, al crescere delle iterazioni, il valore assoluto della differenza fra valore ottenuto e valore atteso di queste funzioni semplici, si ottiene una stima grafica dell'andamento dell'incertezza del n-simo termine di un polinomio.

In questo modo si può ottenere una visualizzazione grafica (4), non rigorosa ma indicativa, del termine di correttezza dello sviluppo della soluzione.

Figura 4: Rappresentazione grafica della dispersione tra valore numerico e valore atteso



Come si vede dai grafici 4, l'errore cresce al crescere dei punti stimati, questo è inevitabile e consegue dalla propagazione dell'errore. Apparentemente l'andamento della crescita è una potenza del ordine del termine meno quello dello del metodo.

La natura progressiva del metodo, che stima un punto basandosi esclusivamente sulle informazioni della soluzione nel punto precedente, fa sì che l'incertezza di ciascun punto risenta non solo della precisione del metodo ma anche del errore sul valore della soluzione nel punto precedente. Ovviamenete ridurre il numero dei punti non aggira il problema, la validità dei metodi di integrazione si fonda sulla possibilità di approssimare una funzione con il suo sviluppo polinomiale troncato e questo è analiticamente ragionevole solo per intervalli sulle ascisse piccoli.

In queste simulazioni è stato usato un incremento di $\tau = 0,05$, i grafici per uno step più piccolo risulterebbero identici ma scalati su errori assoluti minori. In conclusione l'ordine di approssimazione di τ degli algoritmi risulta:

<i>metodo</i>	<i>errore</i>
<i>EULERO</i>	$o(\tau)$
<i>CORRETTORE</i>	$o(\tau^2)$
<i>R – K4</i>	$o(\tau^4)$

Il metodo Runge-Kutta 4 risulta fra tutti il più preciso, ma bisogna ricordare che l'errore che ogni algoritmo può presentare nella stima della n-simo punto della soluzione non dipende solo da esso ma anche dal punto, l'incertezza cresce con il numero di iterazioni.

2.5 Estensione dei metodi R-K ad O.D.E. di ordine superiore al primo

Si consideri il problema di Cauchy di grado n:

$$\begin{cases} \dot{x}^{(n)} &= F(t, x, x', \dots, x^{(n-1)}) \\ x(t_0) &= x_0 \\ \dot{x}(t_0) &= \dot{x}_0 \\ &\vdots \\ x^{(n-1)}(t_0) &= x_0^{(n-1)} \end{cases} \quad (12)$$

Con il cambio di coordinate standard $x^{(j)}(t) = Z_{j+1}(t)$ si può trasformare un'equazione differenziale ordinaria di ordine n in un sistema di n equazioni ordinarie di primo ordine.

$$\begin{cases} \dot{Z}_1 &= Z_2 \\ \dot{Z}_2 &= Z_3 \\ &\vdots \\ \dot{Z}_{n-1} &= Z_n \\ \dot{Z}_n &= F(t, Z_1, \dots, Z_n) \end{cases}$$

Introducendo la notazione vettoriale a n componenti dove la prima colonna del sistema precedente è la derivata prima del vettore $(Z_1(t), \dots, Z_n(t))$ mentre la seconda è la rappresentazione in componenti di una funzione $G(t, \vec{Z}) : R^{n+1} \rightarrow R^n$

si ottiene il problema :

$$\begin{cases} \dot{\vec{Z}} &= G(t, \vec{Z}) \\ \vec{Z}(t_0) &= \vec{x}_0 \end{cases} \quad (13)$$

Che non è altro che un problema di Cauchy di primo ordine con una funzione soluzione di n componenti. $\vec{Z}(t_0)$ è il vettore dei termini noti del problema di ordine n tradotto nella notazione con le funzioni Z_i .

L'estensione di qualsiasi metodo R-K ad equazioni differenziali di grado superiore al primo è simbolicamente ovvio. La forma più generale possibile sarà:

$$\begin{aligned} \vec{K}_0 &= hG(t_0, \vec{Z}(t_0)) \\ \vec{K}_1 &= hG(t_0 + \theta_1 h, \vec{Z}(t_0) + \theta_1 \alpha_1) \\ &\vdots \\ \vec{K}_n &= hG(t_0 + \theta_n h, \vec{Z}(t_0) + \theta_n \alpha_n) \\ \vec{Z}' &= \vec{Z}_0 + \sum_{j=0}^n \omega_j \vec{K}_j \end{aligned} \quad (14)$$

Rendendosi conto che $G(t, \vec{V})$ è semplicemente un operatore che scala di un posto verso l'alto gli elementi del vettore \vec{V} e pone come ultima componente il valore della funzione $F(t, \vec{V})$ e limitandosi al caso semplice dei metodi R-K del tipo (10), dove il coefficiente K_n è dipende esclusivamente dal coefficiente K_{n-1} direttamente precedente, si trova la forma esplicità dei termini K .

$$\vec{K}_0 = h \begin{bmatrix} Z_2(t_0) \\ \vdots \\ Z_n(t_0) \\ F(t_0, Z_1(t_0), \dots, Z_n(t_0)) \end{bmatrix} \quad \vec{K}_n = h \begin{bmatrix} (\vec{Z}(t_0) + \theta_n \vec{K}_{n-1})_2 \\ \vdots \\ (\vec{Z}(t_0) + \theta_n \vec{K}_{n-1})_n \\ F(t_0 + \theta_n h, \vec{Z}(t_0) + \theta_n \vec{K}_{n-1}) \end{bmatrix} \quad (15)$$

Questa non è l'unica possibile estensione a più variabile dei metodi R-K.

Si consideri per semplicità un generico sistema di due equazioni differenziali ordinarie di primo ordine:

$$\begin{cases} \dot{x}(t) = F(t, x(t), u(t)) \\ \dot{u}(t) = G(t, x(t), u(t)) \end{cases} \quad (16)$$

con valori iniziali all'istante iniziale t_n fissati: $x_n = x(t_n)$ e $u_n = u(t_n)$.

A questo punto si può pensare di applicare naturalmente l'algoritmo R-K4 ad ogni equazione differenziale del sistema presa singolarmente. Ovvero si calcolano:

$$\begin{aligned} x_{n+1} &= x_n + (k_0 + 2k_1 + 2k_2 + k_3)/6 \\ u_{n+1} &= u_n + (m_0 + 2m_1 + 2m_2 + m_3)/6 \end{aligned} \quad (17)$$

Dove:

$$\begin{aligned} k_0 &= hF(t_n, x_n, u_n) & m_0 &= hG(t_n, x_n, u_n) \\ k_1 &= hF(t_n + h/2, x_n + k_0/2, u_n + m_0/2) & m_1 &= hG(t_n + h/2, x_n + k_0/2, u_n + m_0/2) \\ k_2 &= hF(t_n + h/2, x_n + k_1/2, u_n + m_1/2) & m_2 &= hG(t_n + h/2, x_n + k_1/2, u_n + m_1/2) \\ k_3 &= hF(t_n + h, x_n + k_2, u_n + m_2) & m_3 &= hG(t_n + h, x_n + k_2, u_n + m_2) \end{aligned}$$

Questa formulazione è un'estensione abbastanza intuitiva del algoritmo fatto per una singola variabile, il primo passo è il calcolo della derivata prima di x e di u nel punto iniziale t_0 , da queste si calcolano k_0 e m_0 che sono una prima stima dell'incremento finito delle due soluzioni $x(t)$ e $u(t)$ dopo lo step h .

Questi due coefficienti vengono poi utilizzati per ottenere una prima stima delle derivate delle soluzioni (passando chiaramente attraverso le funzioni F e G) nel punto medio e così via esattamente come si è costruito il metodo per l'equazione singola.

In modo analogo a quanto fatto prima è possibile vedere un'equazione differenziale di secondo grado come un sistema di 2 equazioni, definendo $(x)(t) = u(t)$, basterà porre nel sistema di equazioni (11) $F(t, x(t), u(t)) = u(t)$ per ottenere l'algoritmo corrispondente a questo tipo di problemi. A questo punto l'implementazione in C++ è praticamente ovvia:

```
void RK4(double (*G)(double, double, double), double *p0, double h) //←
    ingresso: funzione, valori iniziali, incremento
{
    double x=p0[0];
    double y=p0[1];
    double u=p0[2];
    double k[4], m[4];

    k[0] = h * u;
    m[0] = h * G(x, y, u);
```

```

k[1] = h * (u+ m[0]/2);
m[1] = h * G(x + h/2 , y +k[0]/2 , u +m[0]/2);
k[2] = h * (u+ m[1]/2);
m[2] = h * G(x + h/2, y + k[1]/2 , u + m[1]/2);
k[3] = h * (u + m[2]);
m[3] = h * G(x+h,y + k[2] , u + m[2]);

p0[0] = x +h;
p0[1] = y + (k[0]+2*k[1]+2*k[2]+k[3])/6;
p0[2] = u + (m[0] + 2*m[1] + 2*m[2] +m[3])/6;
}

```

3 Casestudy

Pendolo Caotico E' il sistema fisico formato da un pendolo rigido, di lunghezza R e massa M , vincolato a muoversi su un piano verticale (rotatore rigido piano) su cui incidono 3 forze:

- Forza peso: modulo $F_g = mg$, direzione verticale.
- Forza d'attrito : $\vec{F}_\eta = -\eta \vec{v}$, stessa direzione ma verso opposto alla velocità e di modulo proporzionale ad un parametro η .
- Forzante armonica: la direzione ruota con velocità angolare fissa ω mentre il modulo $F_c = F_0$ è costante.

Come si può vedere nella figura 3.

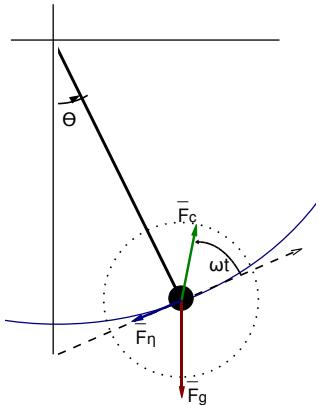


Figura 5: Modello di Pendolo Caotico

equazioni del moto Il sistema ha solo un grado di libertà, scegliendo come coordinata di configurazione l'angolo θ sotteso dalla verticale e dalla direzione del rotatore, si ricava l'equazione di newton del sistema:

$$\begin{aligned} I \ddot{\theta} &= \sum \tau = \\ &= m R^2 \ddot{\theta} = R(F_0 \cos(\omega t) - mg \sin(\theta) - \eta R \dot{\theta}) \end{aligned} \quad (18)$$

che conduce alla seguente O.D.E. di secondo grado

$$\ddot{\theta} = \frac{F_0}{mR} \cos(\omega t) - \frac{g}{R} \sin(\theta) - \eta R \dot{\theta} \quad (19)$$

Scegliendo per semplicità un pendolo di massa 1 kg e lunghezza $9,8 \text{ m}$ (in altre parole la costante moltiplicativa $B = \frac{g}{R}$ del seno viene fissata ad 1s^{-2}) rimangono da fissare a seconda dei casi le costanti che definiscono la struttura del sistema

$$A = \frac{\eta}{m} \quad \Phi = \frac{F_0}{mR} \quad \omega$$

e le condizioni al iniziali $\theta(t_0)$ e $\dot{\theta}(t_0)$.

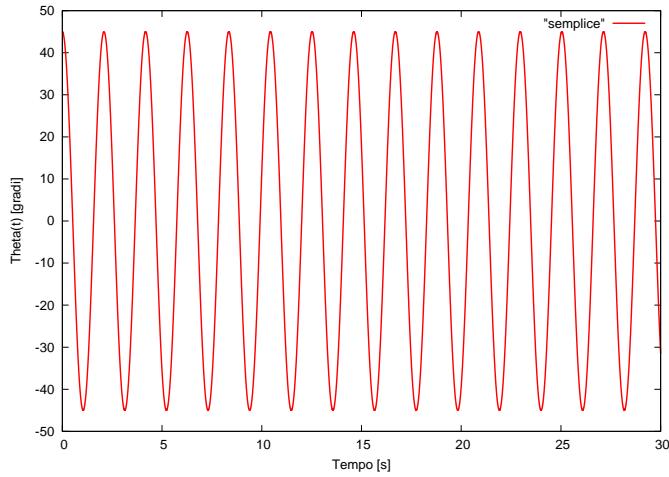
4 Risultati

4.1 Applicazione R-K4 a casi banali

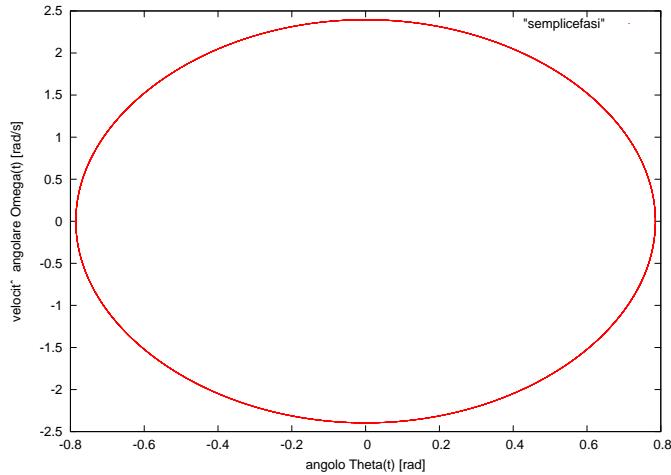
Studio della dinamica del sistema risolvendo numericamente l'equazione del moto.

Figura 6

Pendolo semplice Dati: $A = 0$; $\Phi = 0$; $\theta(t_0) = \pi/4$



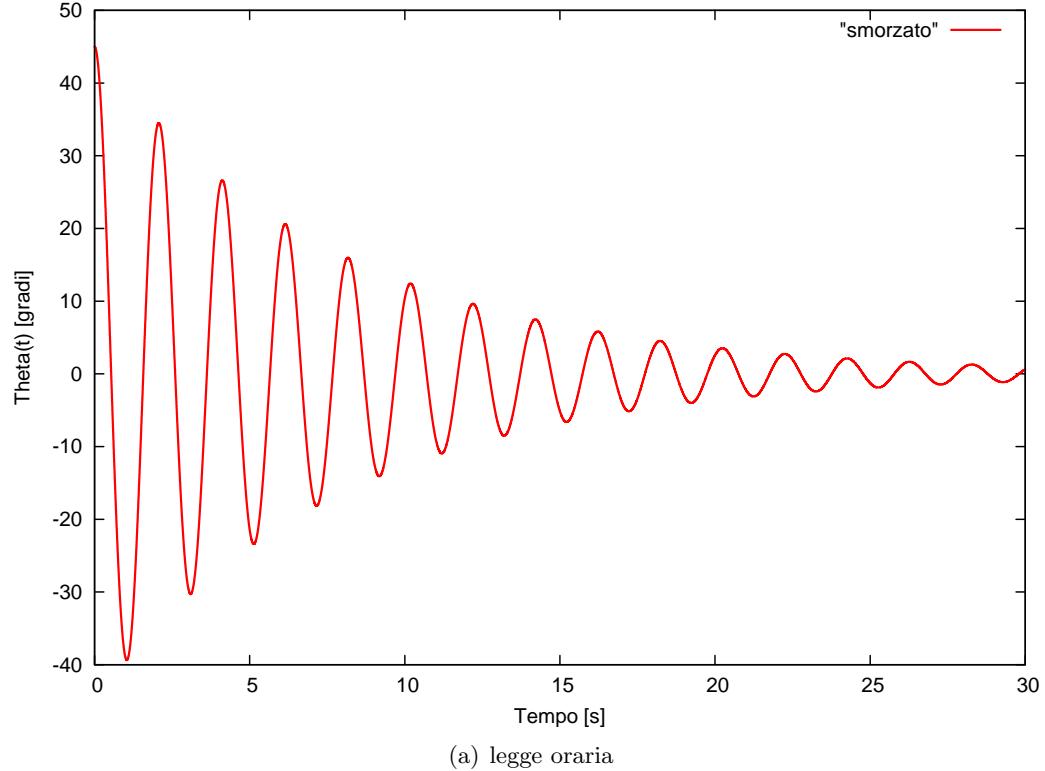
(a) legge oraria



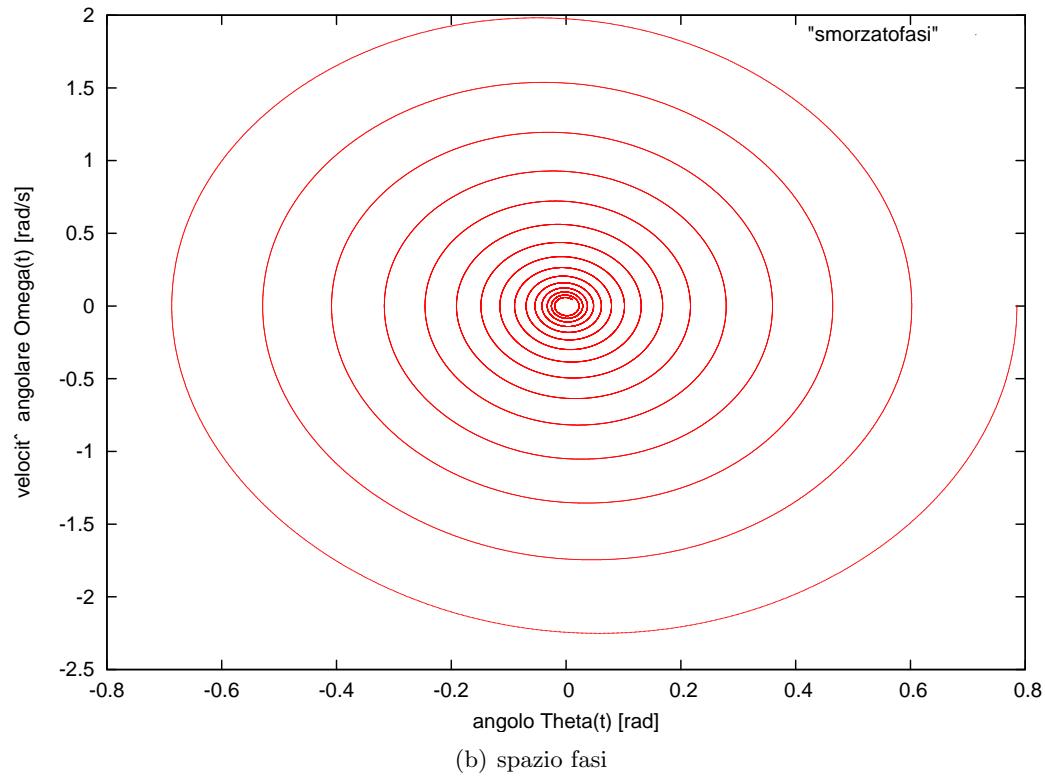
(b) spazio fasi

Figura 7

Pendolo smorzato Dati: $A = 0,25 \text{ s}^{-1}$; $\Phi = 0$; $\theta(t_0) = \pi/4$



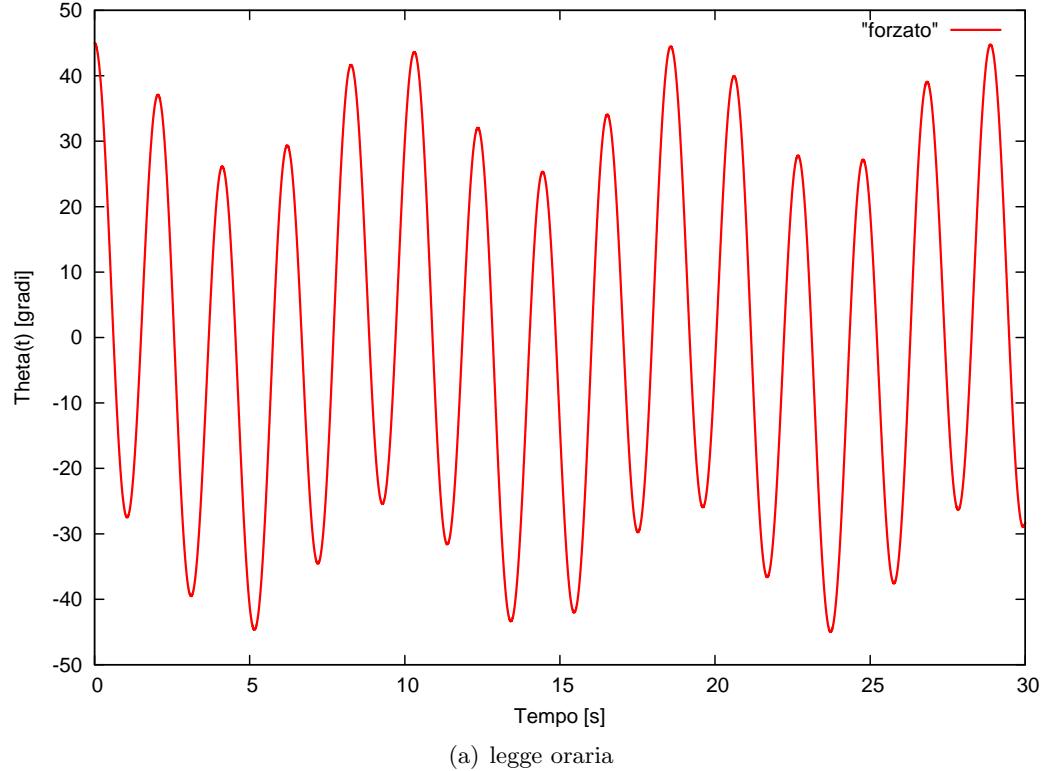
(a) legge oraria



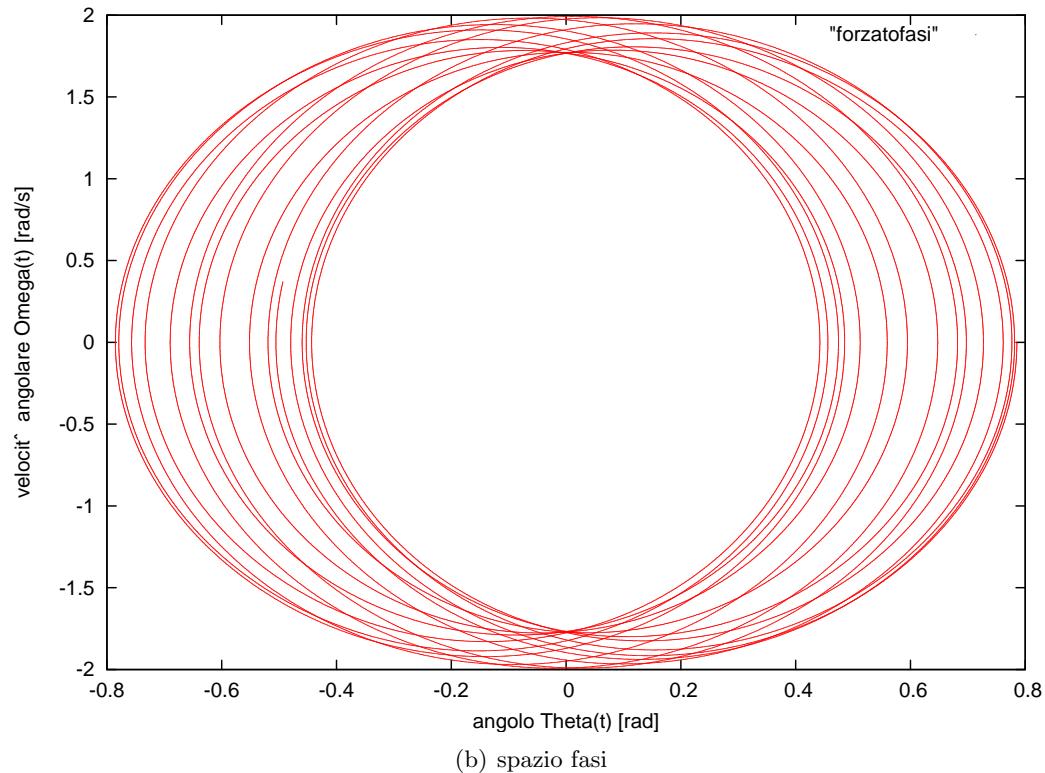
(b) spazio fasi

Figura 8

Pendolo Forzato Dati: $A = 0$; $\Phi = 1,5 \text{ s}^{-2}$; $\omega = 0,66 \text{ s}^{-1}$; $\theta(t_0) = \pi/4$



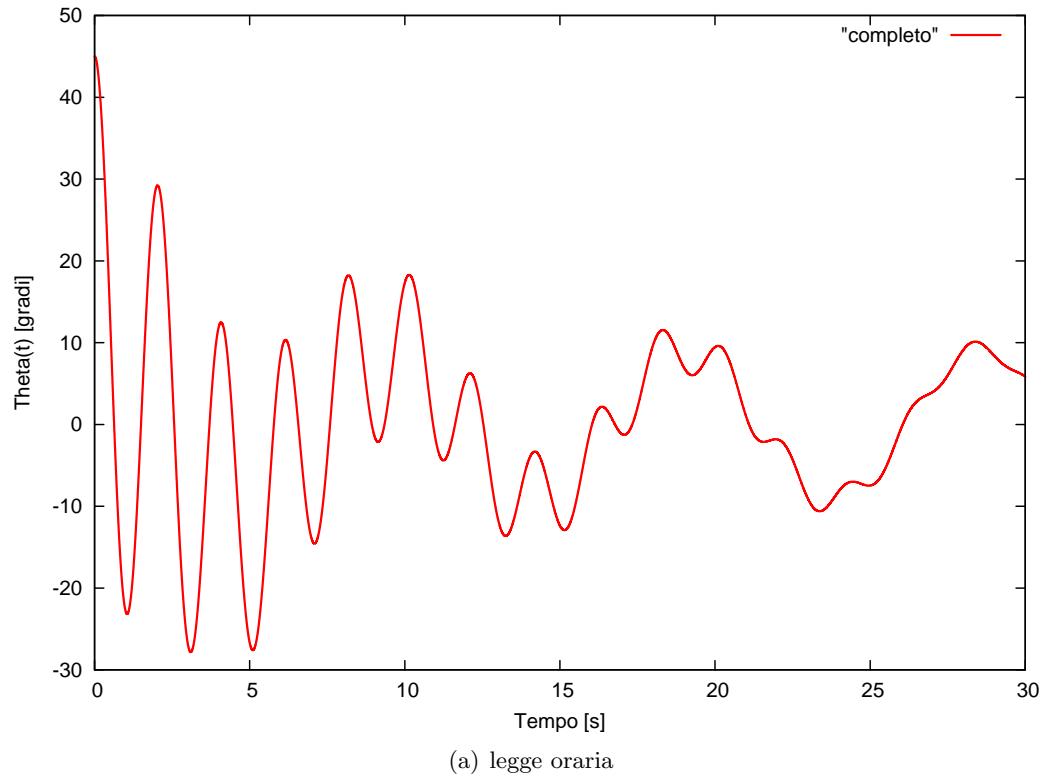
(a) legge oraria



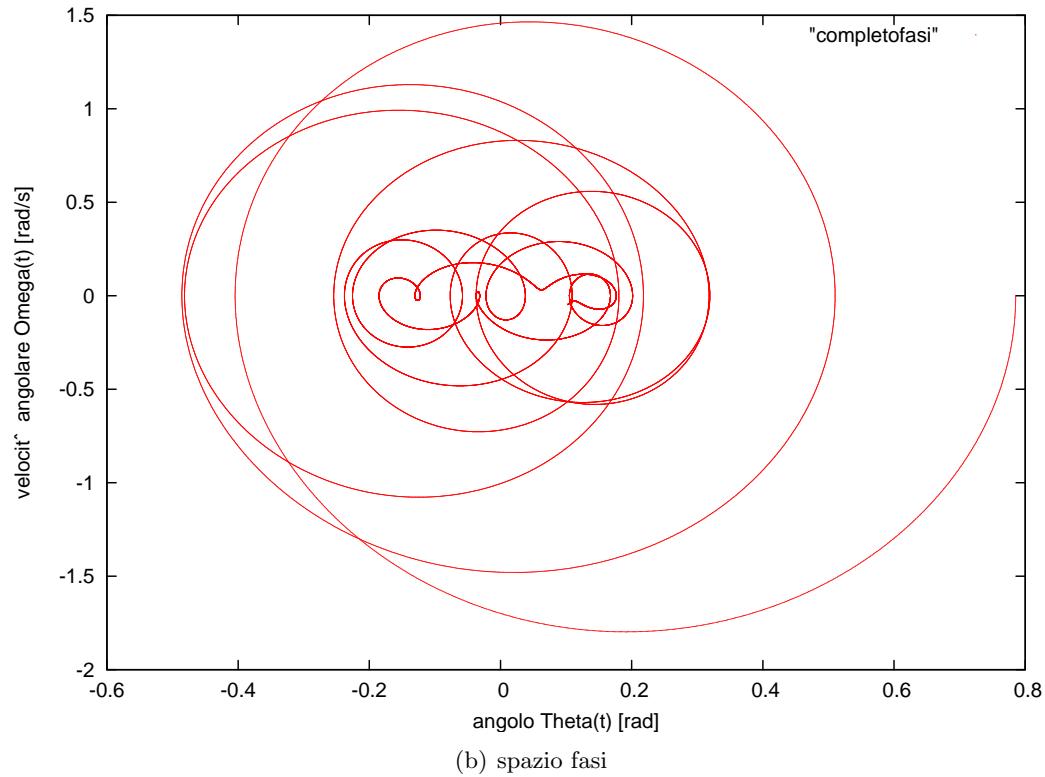
(b) spazio fasi

Figura 9

Pendolo Caotico Dati: $A = 0,25 \text{ s}^{-1}$; $\Phi = 1,5 \text{ s}^{-2}$; $\omega = 0,66 \text{ s}^{-1}$;
 $\theta(t_0) = \pi/4$



(a) legge oraria



(b) spazio fasi

4.2 Caoticità del Sistema

Conoscendo esattamente l'equazioni del moto l'evoluzione del sistema è perfettamente deterministica. Nonostante questa c'è forte dipendenza del evoluzione del sistema dai parametri iniziali, in altre parole una variazione infinitesima dei coefficienti strutturali (η , Φ , ω , θ_0) del sistema attorno a particolari valori critici determinano una notevole divergenza tra le due traiettorie corrispondenti alle diverse configurazioni.

Questo fenomeno è detto Pseudo-Caoticità del sistema: due istanze di un sistema che partono con condizioni iniziali molto simili si separano molto rapidamente, l'evoluzione dipende molto sensibilmente dalle condizioni iniziali.

Individuazione delle configurazioni caotiche La prima cosa che si può dimostrare sperimentalmente è che il sistema non ha comportamento caotico per ogni configurazione. Si studiano numericamente le traiettorie del sistema in 3 configurazioni che differiscono solo per il valore del parametro Φ .

Parametri comuni:

$$A = 0,5 \quad \omega = 0,66\text{s}^{-1} \quad \theta_0 = 0 \quad \dot{\theta}_0 = 0$$

Parametri variati:

$F0_{\text{alta}}$	$\Phi = 2,150\text{s}^{-2}$
$F0_{\text{caos}}$	$\Phi = 1,48555\text{s}^{-2}$
$F0_{\text{bassa}}$	$\Phi = 0,150\text{s}^{-2}$

Nella figura 10 è possibile vedere a confronto le tre traiettorie, nel grafico è evidente la peridicità delle due configurazioni limite a dispetto della apparente casualità della soluzione intermedia, ci si aspetterà che i parametri della configurazione $F0_{\text{caos}}$ siano nell'intorno della configurazione caotica.

Nelle figure 11 e 12 si prende invece in esame il comportamento delle configurazioni estreme per piccole variazioni del parametro Φ .

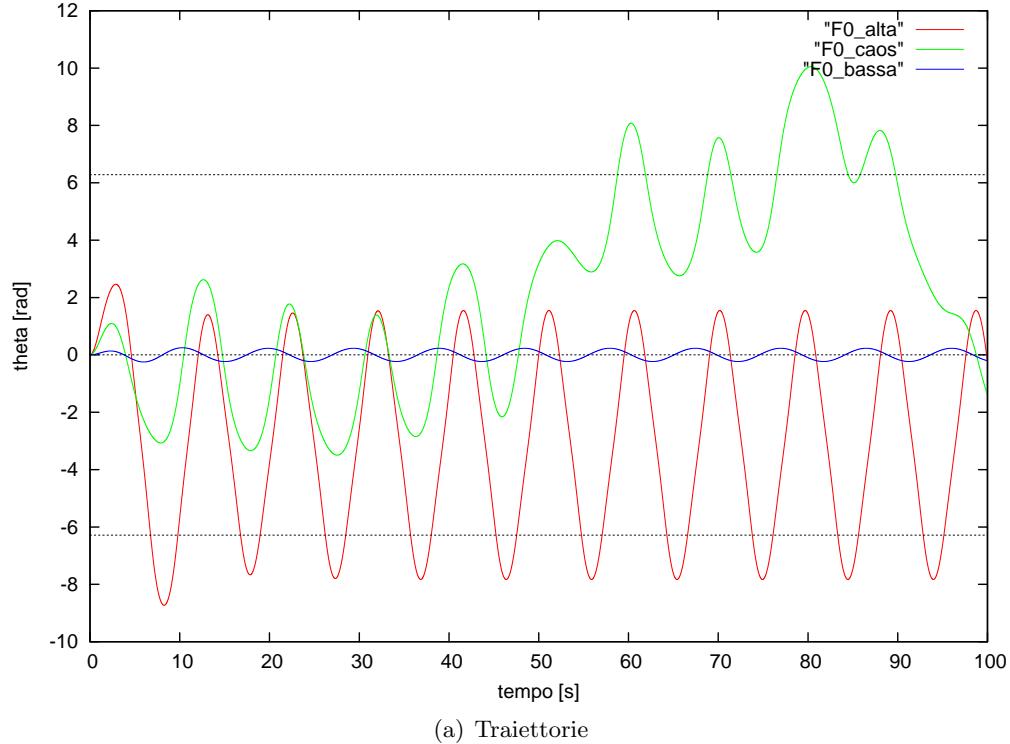
Si nota subito che nell'intorno di questi valori le traiettorie variano in modo continuo per piccole variazioni ($\pm 0,005\text{s}^{-2}$) del parametro A .

In queste configurazioni il sistema non mostra quindi un comportamento pseudo-caotico, sono in un certo senso in equilibrio stabile vicino alle configurazioni $F0_{\text{alta}}$ e $F0_{\text{bassa}}$.

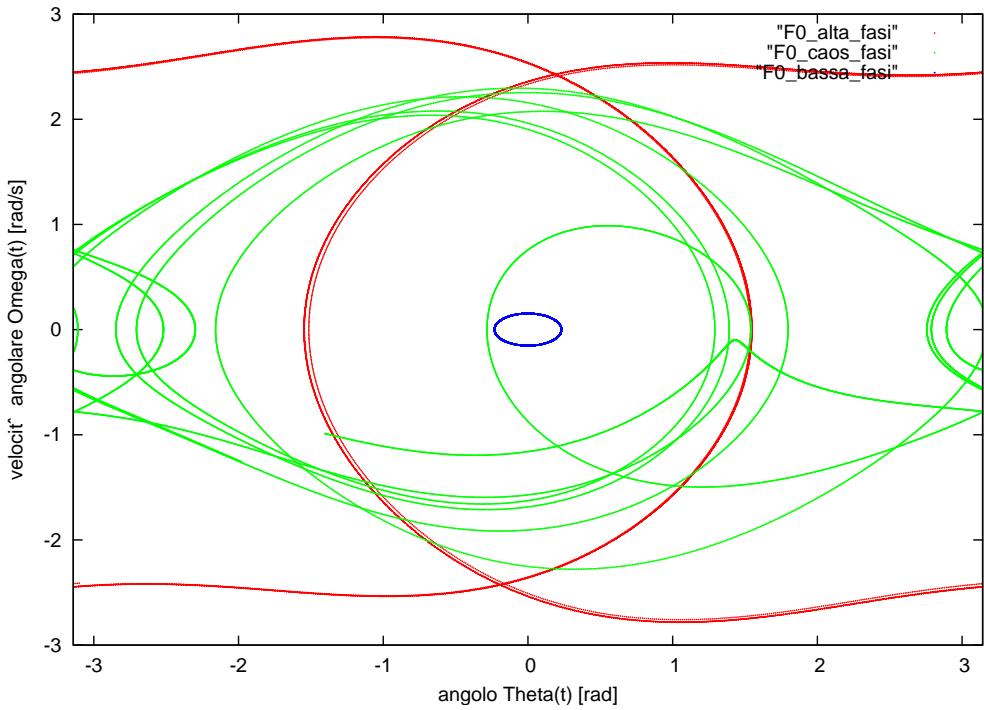
(oss: per tutti i grafici relativi agli spazi delle fasi è stato trascurato il transiente iniziale)

Illustrazione del comportamento caotico Individuata approssimativamente la regione di instabilità delle soluzioni attorno ai parametri di $F0_{\text{caos}}$ si può mostrare il senso della pseudocaoticità costruendo il grafico di altri 2 sistemi i cui parametri differiscono infinitesimalmente da quella caotica: grafici 13, 14 e 15. I sistemi che si trovano in questa regione di parametri avranno una traiettoria che dopo un tempo finito biforcherà da quella infinitesimalmente vicina.

Figura 10

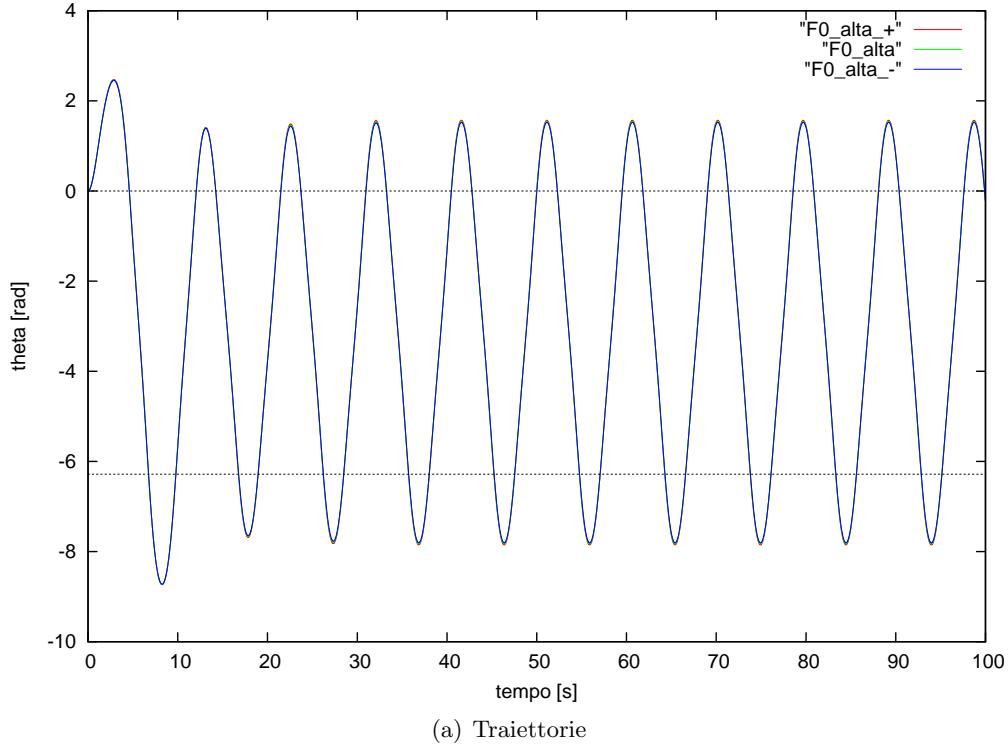
Confronto dei 3 pendoli con diverse forzanti

(a) Traiettorie

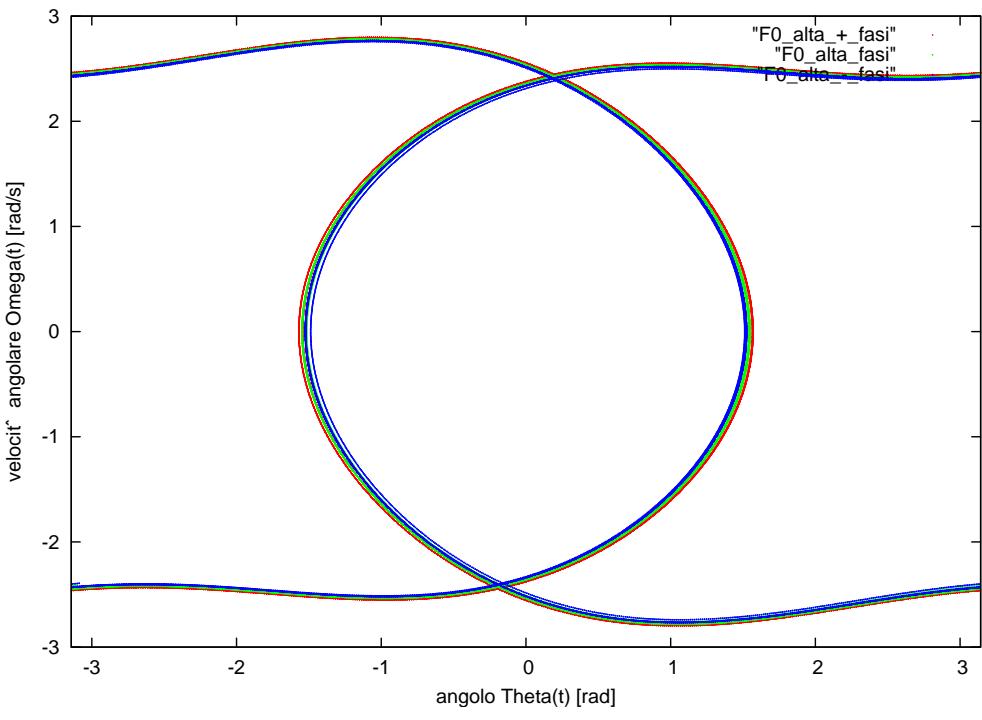


(b) Spazio delle Fasi

Figura 11

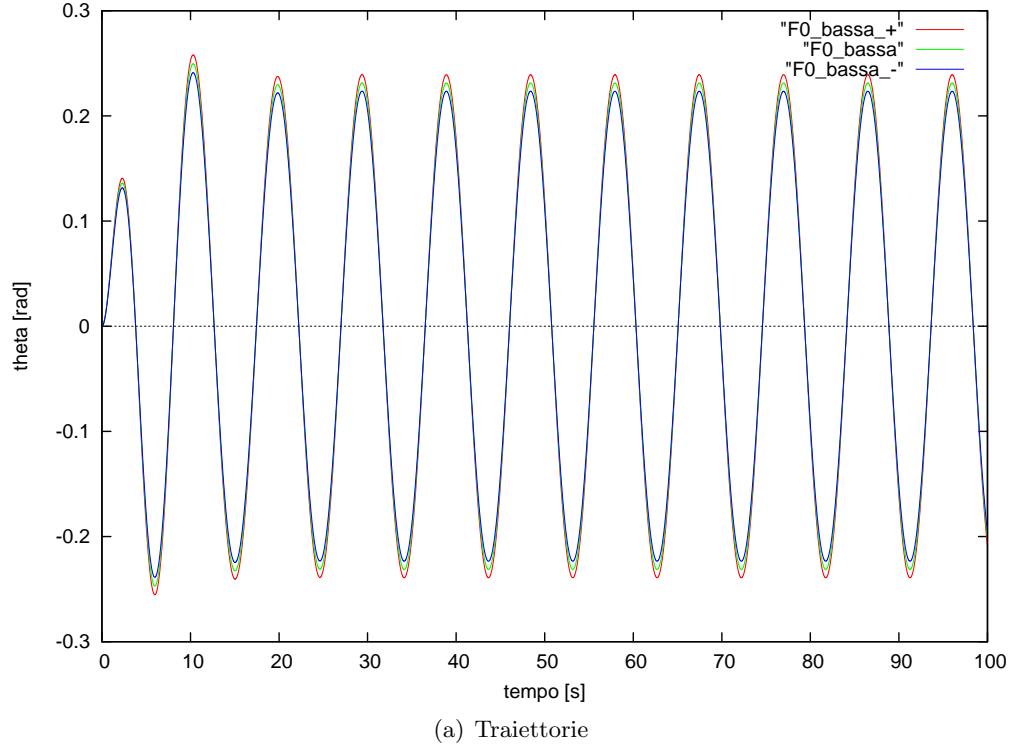
Confronto fra 3 configurazioni intorno ad F0_alta

(a) Traiettorie

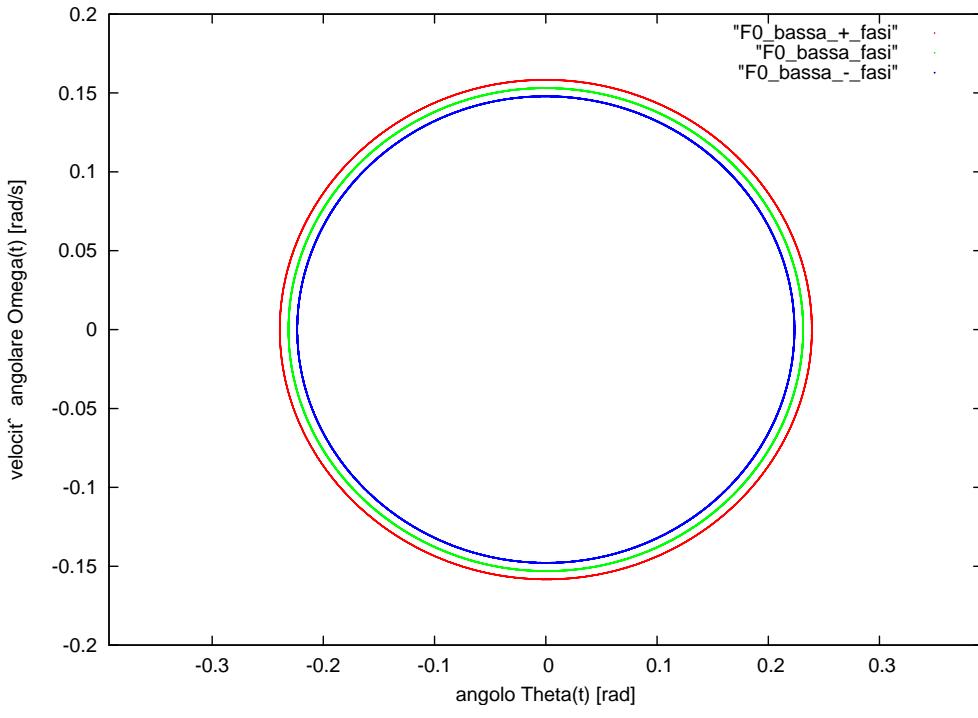


(b) Spazio delle Fasi

Figura 12

Confronto fra 3 configurazioni intorno ad F0_bassa

(a) Traiettorie



(b) Spazio delle Fasi

Figura 13

3 traiettorie di pendolo con angolo iniziale θ variato di ± 0.005 rad attorno ad F0_caos

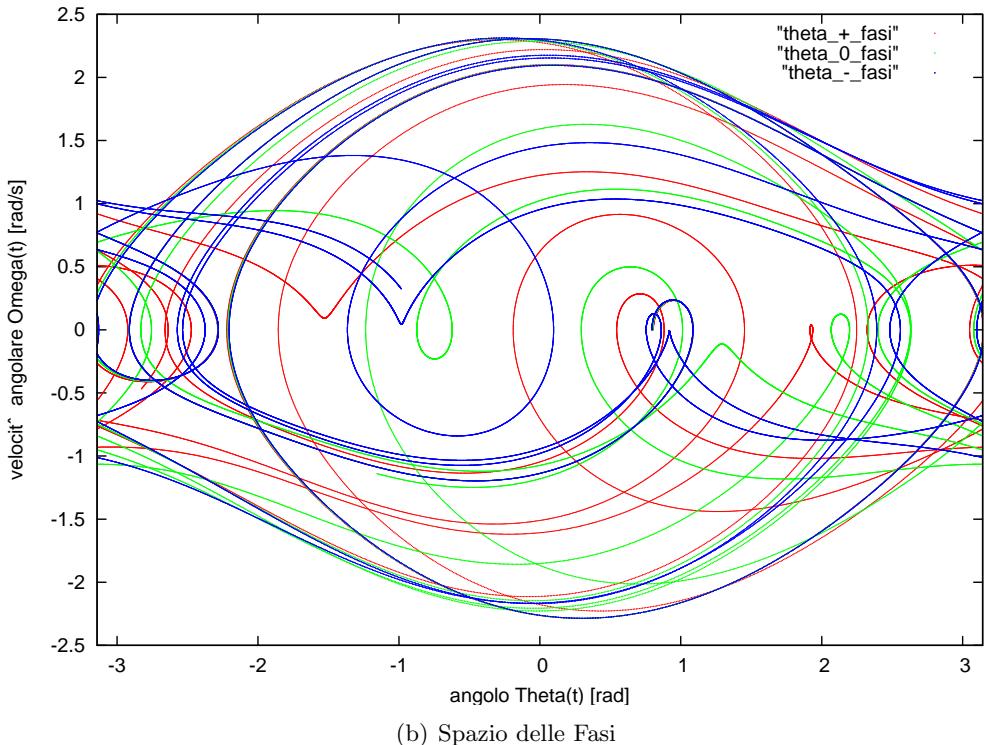
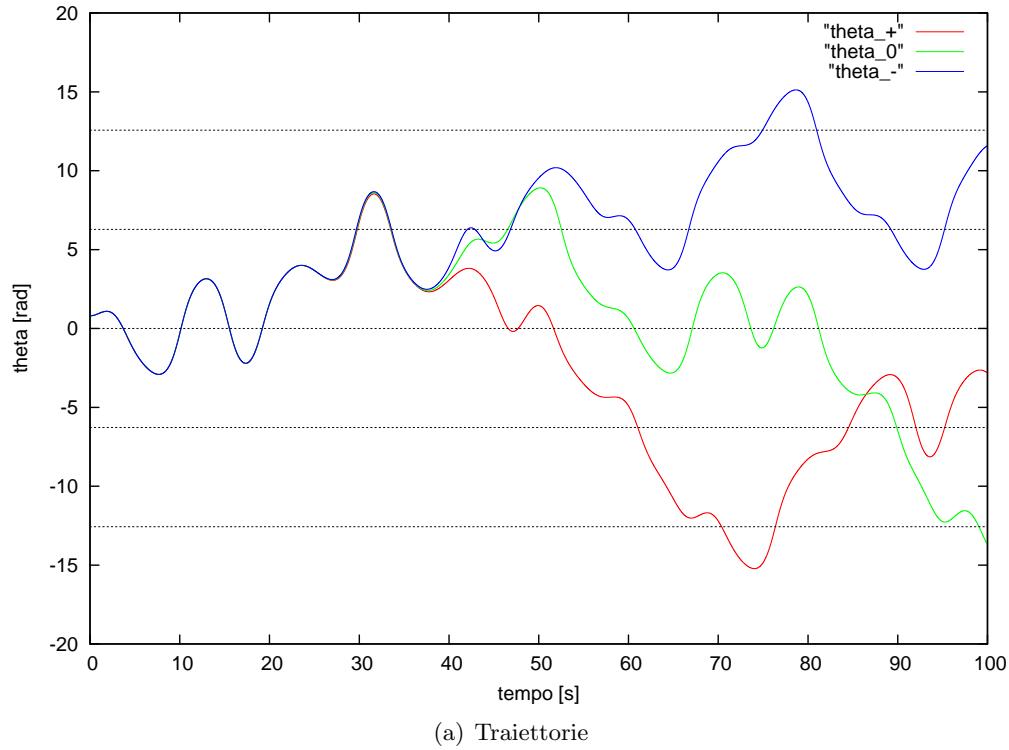
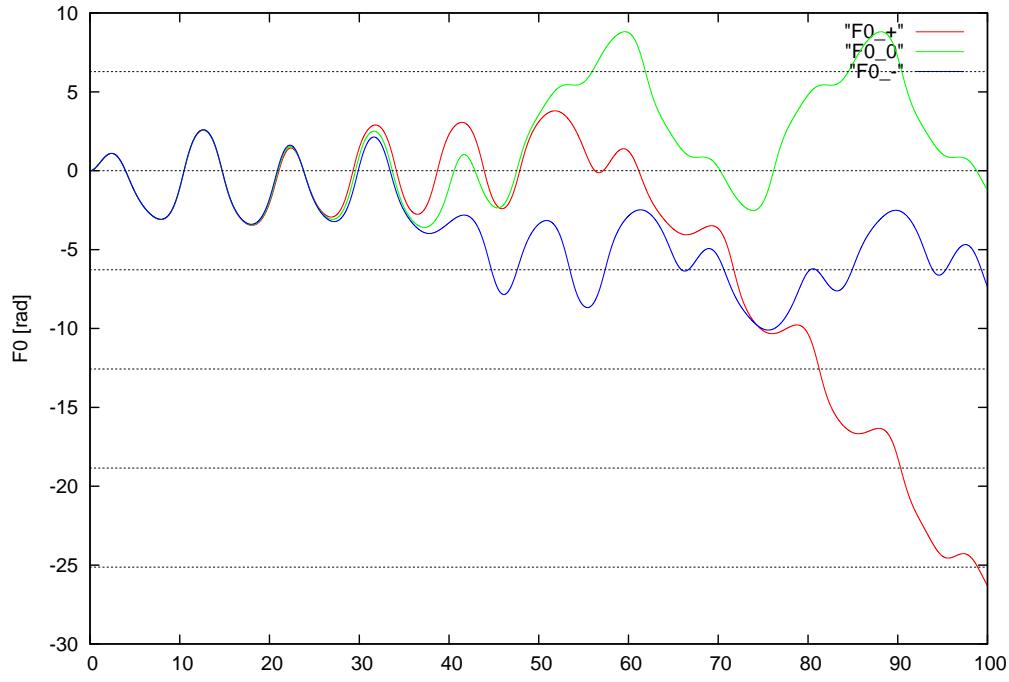
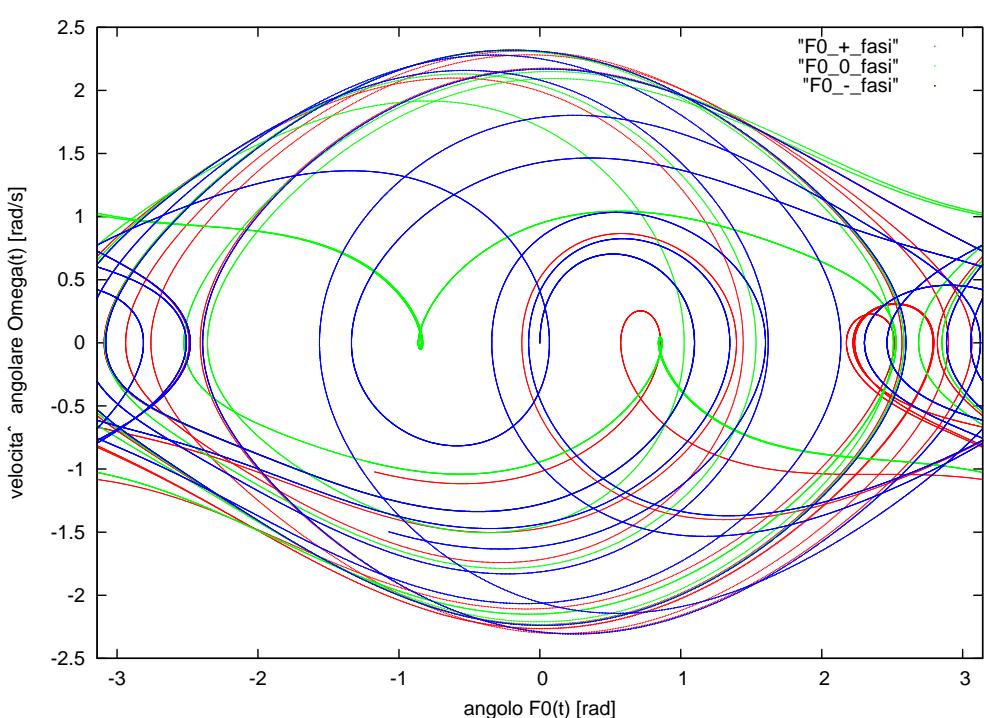


Figura 14

traiettorie di 3 pendoli con forzanti variate di $\pm 0.005 s^{-2}$ rispetto al valore di F0_caos



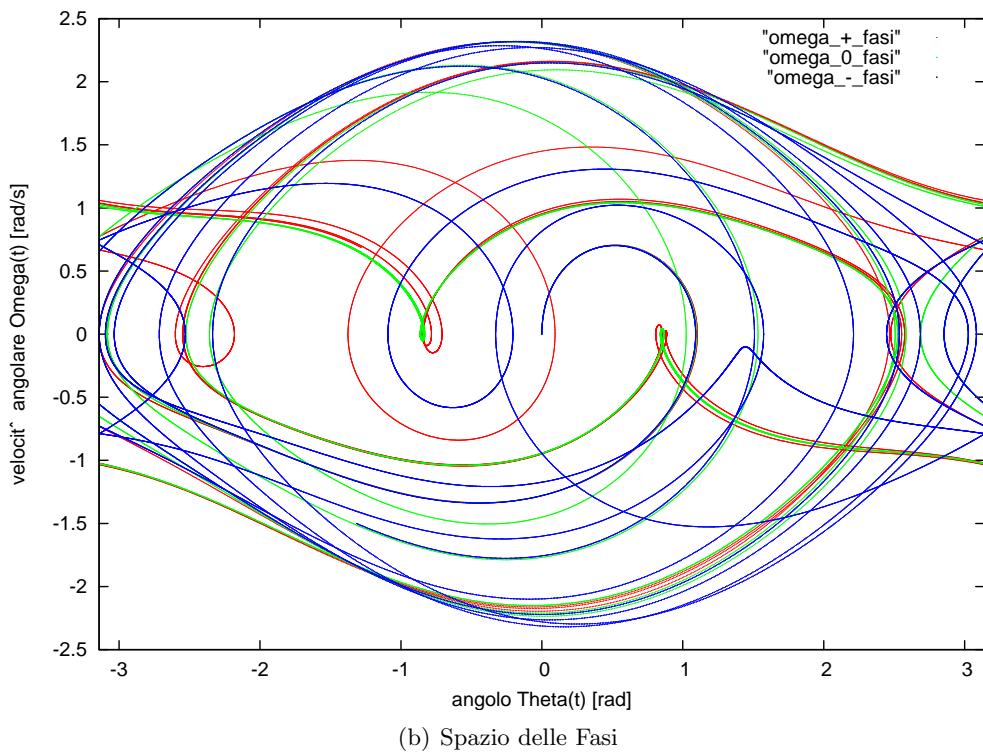
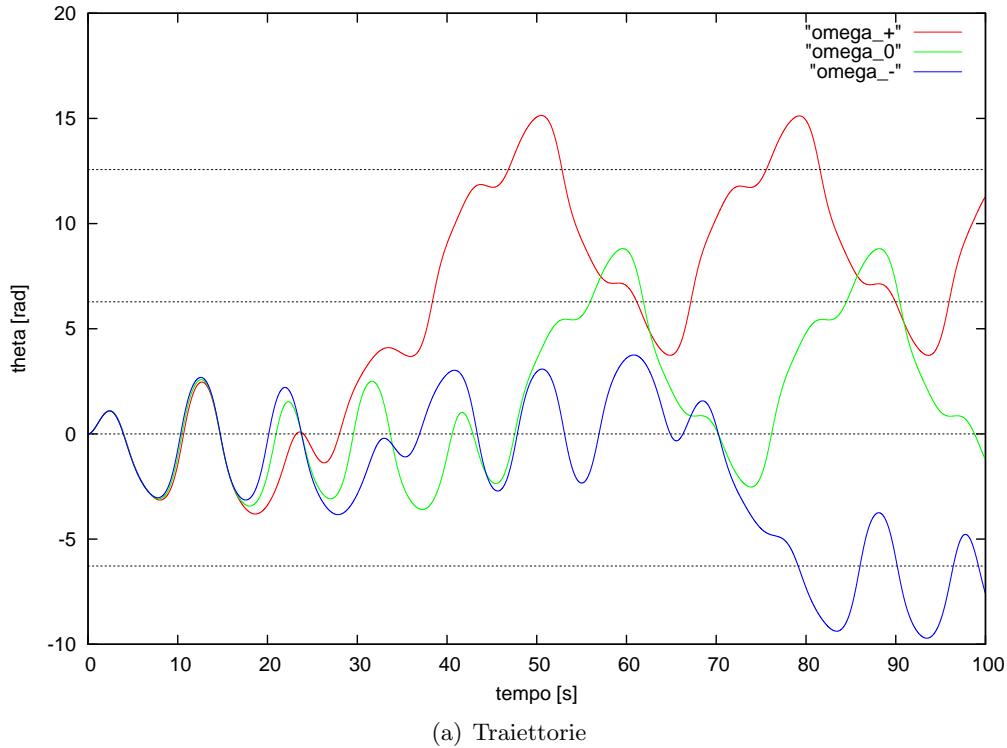
(a) Traiettorie



(b) Spazio delle Fasi

Figura 15

traiettorie di 3 pendoli con frequenza delle forzanti variate di $\pm 0.00005 s^{-2}$ rispetto al valore di F0_caos



Riferimenti bibliografici

- [1] *Numerical recipes in C++*
- [2] Knuth D. *The Art of Computer Programming*
- [3] Hildebrand F. *Introduction to Numerical Analysis*, 2nd.ed.