

Simulazione delle sezioni di Poincaré di un modello di Hénon e Heiles

Fabio Peluso

Sotto la supervisione del Prof. Ugo Locatelli
nell'ambito del progetto INCA-ABACO

16 giugno 2017

1 Framework

1.1 Sezione di Poincaré

La sezione di Poincaré è un utile metodo per investigare le proprietà di un sistema dinamico. Sia data l'equazione differenziale autonoma in \mathbb{R}^n

$$\dot{x} = f(x)$$

e sia $\Sigma \subset \mathbb{R}^n$ una ipersuperficie di dimensione $n-1$ ortogonale al flusso $\Phi_t(x_0)$ con $x_0 \in \Sigma$ condizione iniziale del sistema. Si definisce sezione di Poincaré l'applicazione

$$\Sigma \ni x_i \xrightarrow{\Pi} \Pi(x_i) = \Phi_{T_i}(x_0) \in \Sigma$$

dove T_i è il tempo di ritorno i -esimo del flusso partito da x_0 (e.g. T_1 è il tempo del primo ritorno del flusso su Σ), con la condizione aggiuntiva che il verso della derivata $\dot{x}(T_i) = f(x_i)$ è di segno concorde con quello iniziale $\dot{x}(0) = f(x_0)$ rispetto alla normale all'ipersuperficie Σ .

1.2 Modello di Hénon e Heiles

Il modello di Hénon e Heiles è descritto dall'Hamiltoniana

$$\mathcal{H}(\underline{p}, \underline{q}) = \frac{\omega_1}{2}(p_1^2 + q_1^2) + \frac{\omega_2}{2}(p_2^2 + q_2^2) + q_1^2 q_2 - \frac{1}{3} q_2^3 \quad .$$

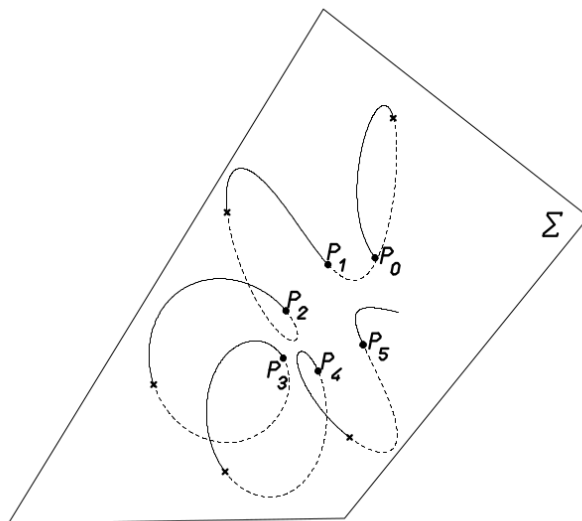


Figura 1: Rappresentazione schematica della sezione di Poincaré. Con riferimento alla definizione nel testo, si ha che $\Pi(P_0) = P_1$, $\Pi(P_1) = P_2$, ... Questa figura è inclusa nelle presenti note su gentile concessione del Prof. A. Giorgilli.

Siamo nel caso $n = 4$; fissando il valore dell'energia possiamo ottenere delle superfici di dimensione 3, al fine di studiare successivamente le sezioni di Poincaré del modello. Facciamo qualche osservazione sul comportamento di questo modello:

- per valori piccoli dell'energia, $\|q\| \ll 1$, il sistema si comporta come una coppia di oscillatori armonici con frequenze $\omega_1 > 0$, $\omega_2 > 0$ con una piccola perturbazione costituita dai termini cubici;
- la superficie Σ individuata dall'equazione $q_1 = 0$ è trasversa al flusso se l'energia è positiva, come segue immediatamente dalle equazioni di Hamilton:

$$\dot{q}_1 = \frac{\partial \mathcal{H}}{\partial p_1} \quad \dot{q}_2 = \frac{\partial \mathcal{H}}{\partial p_2} \quad \dot{p}_1 = -\frac{\partial \mathcal{H}}{\partial q_1} \quad \dot{p}_2 = -\frac{\partial \mathcal{H}}{\partial q_2};$$

- fissato il valore E dell'energia, le intersezioni del flusso con la superficie Σ giacciono sul sottospazio bidimensionale dato dall'equazione

$$\mathcal{H}(p_1, p_2, 0, q_2) = \frac{\omega_1}{2} p_1^2 + \frac{\omega_2}{2} (p_2^2 + q_2^2) - \frac{1}{3} q_2^3 = E$$

e deduciamo che un punto (p_2, q_2) delle sezioni di Poincaré individua in modo univoco un'orbita dal momento che abbiamo $q_1 = 0$ e p_1 dato dall'equazione precedente assumendo che il segno sia positivo;

- il potenziale associato al modello è dato da

$$V(q_1, q_2) = \frac{\omega_1}{2}q_1^2 + \frac{\omega_2}{2}q_2^2 + q_1^2q_2 - \frac{1}{3}q_2^3;$$

lo studio di tale potenziale mostra che la sezione di Poincaré è limitata alla regione individuata dall'equazione

$$\frac{\omega_2}{2}(p_2^2 + q_2^2) - \frac{1}{3}q_2^3 \leq E$$

purché E sia minore della velocità di fuga individuata da

$$E_f = \min \left\{ \frac{\omega_1^3}{24} + \frac{\omega_1^2\omega_2}{8}, \frac{\omega_2^3}{6} \right\}.$$

- il modello è un sistema reversibile, in quanto autonomo e pari nei momenti: infatti se $(\underline{p}(t), \underline{q}(t))$ è una soluzione, lo è anche la sua coniugata temporale data da $(\underline{\tilde{p}}(t), \underline{\tilde{q}}(t)) = (-\underline{p}(-t), \underline{q}(-t))$.

2 Simulazioni numeriche

Lo studio delle sezioni di Poincaré del modello è stato effettuato utilizzando un mini-cluster costituito da 15 Raspberry Pi [®], la cui descrizione dettagliata è disponibile alla pagina web http://160.80.216.190/~inca-abaco/hardware.html#mini-cluster_foto+schema.

Utilizzando tale infrastruttura, è stato possibile eseguire in parallelo un programma scritto in linguaggio C, che calcola le sezioni di Poincaré per 15 valori diversi delle condizioni iniziali, a fissato livello di energia $E = 0.039344$.

Il suddetto programma (che si chiama `omp_henon-heiles-sez-poin.c`), il file che definisce i valori iniziali di ω_1 , ω_2 e delle condizioni iniziali, insieme a tutti gli script di comandi che permettono di eseguire il programma in parallelo su tutti i Raspberry sono compressi in `omp_henon-heiles.zip`, disponibile per il download al link http://160.80.216.190/~inca-abaco/tutorials/omp_henon-heiles.zip. Lo schema di esecuzione parallela è analogo a quello discusso nell'esempio banale di programmazione parallela sul nostro mini-cluster, che è descritto in http://160.80.216.190/~inca-abaco/tutorials/mpi_basics.html

L'algoritmo di integrazione utilizzato è il metodo SBAB3C introdotto da Laskar e Robutel in [5]. A sua volta il metodo SBAB3C può essere visto come una evoluzione di quello del punto centrale, o Leap-Frog, che è accuratamente descritto in [3].

Le seguenti figure mostrano rispettivamente le condizioni iniziali del sistema su cui è stata fatta l'integrazione e le sezioni di Poincaré nello spazio delle fasi (q_2, p_2) .

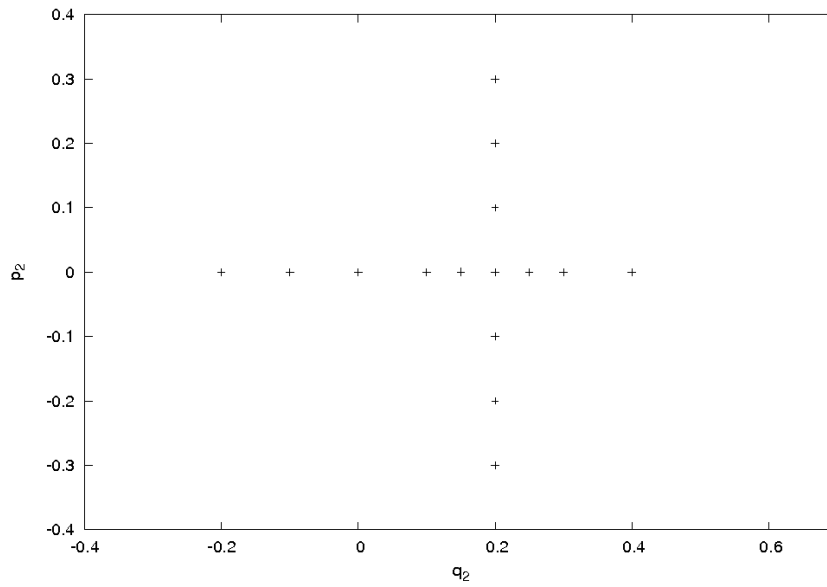


Figura 2: Condizioni iniziali

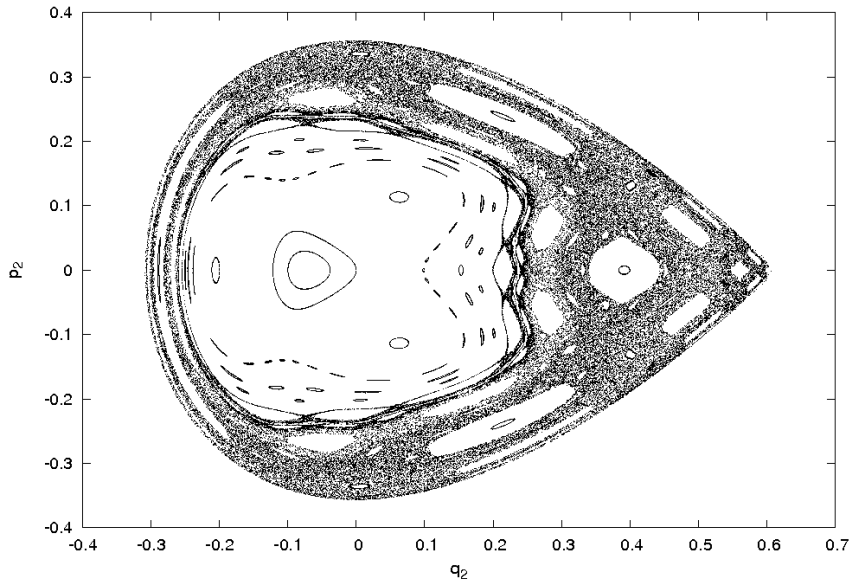


Figura 3: Sezione di Poincaré per il caso non risonante ($\omega_1 = 1$ e $\omega_2 = (\sqrt{5} - 1)/2$) del modello di Hénon e Heiles corrispondente al valore dell'energia $E = 0.039344$, che è leggermente inferiore a quello di fuga.

3 Conclusioni

Il caso in esame, che è non risonante (cioè tale che $\nexists(k_1, k_2) \in \mathbb{Z}^2 \setminus (0, 0)$ per cui si ha che $k_1\omega_1 + k_2\omega_2 = 0$), mostra una zona centrale con curve chiuse invarianti concentriche. Sono presenti delle zone in cui è possibile il moto ordinato su curve chiuse 1D che stanno intorno ad orbite periodiche. Tuttavia è molto ampia l'area della superficie che viene riempita in maniera apparentemente erratica e abbastanza uniforme dall'orbita che parte da un punto iniziale all'interno di quella stessa regione. Questo comportamento è tipico delle orbite caotiche (cfr. [2])

Lo scenario è quindi il medesimo rispetto a quello descritto in [6], dove troviamo anche un'analisi per diversi livelli di energia e una breve discussione del caso risonante.

4 Breve analisi delle prestazioni del nostro mini-cluster

Per comprendere i vantaggi dell'uso del mini-cluster, il programma è stato eseguito anche su Marvin, cioè il server/workstation principale del nostro cluster. I tempi di esecuzione riportati nel seguito sono stati ottenuti utilizzando il comando `time`.

Per comprendere meglio le differenze tra i due apparati, è bene ricordare che il mini-cluster è composto da 15 Raspberry Pi 3, ciascuno equipaggiato con un processore quad-core da 1.2GHz a 64bit con architettura ARM, mentre Marvin monta un processore Intel i5 quad-core da 3.3 GHz a 64bit.

Il tempo che corrisponde al parametro `real` è il tempo totale per l'esecuzione del programma; in esso si tiene conto anche dei tempi di lettura e scrittura da file, e di passaggio di informazioni tra Raspberry e Marvin; dunque dipende anche dalla velocità della rete locale. Il tempo corrispondente a `user` è il tempo effettivo di utilizzo della CPU per il calcolo. Per maggiori dettagli sulle specifiche di questi tempi si rimanda ai manuali tecnici.

Per studiare al meglio i diversi comportamenti di Marvin e del mini-cluster il programma è stato eseguito prima con tempo totale di integrazione numerica delle equazioni del moto $T_F = 10^5$ e poi con tempo $T_F = 10^6$.

Analizzando i tempi di esecuzione per $T_F = 10^5$ si nota che il tempo totale di esecuzione (`real`) è maggiore per i Raspberry di circa 21 secondi, il tempo di calcolo effettivo stimato non presenta grandi differenze. È molto interessante notare che su Marvin il tempo totale di esecuzione e il tempo effettivo di calcolo differiscono per meno di un secondo, mentre la differenza è molto più marcata per il mini-cluster.

Il fatto che il tempo complessivo (`real`) di esecuzione sia maggiore per i Raspberry è quasi certamente dovuto alla necessità di inviare e ricevere le informazioni tramite rete locale, con il passaggio di numerosi file tra Marvin e i Raspberry. Quando il programma viene eseguito solo su Marvin non è necessario trasferire alcun file, con un evidente risparmio di tempo.

Per quanto riguarda il tempo effettivo di calcolo, il mini-cluster è in leggero vantaggio e se ne può concludere che la parallelizzazione del problema ottiene l'effetto sperato di ridurre il tempo di calcolo effettivo.

Il secondo test, effettuato con tempo di integrazione $T_F = 10^6$, mette maggiormente in evidenza il vantaggio di utilizzare un mini-cluster per problemi come il nostro, che ben si prestano alla parallelizzazione. Infatti in questo secondo test il mini-cluster batte Marvin sul tempo totale di esecuzione. Analizzando i tempi, complessivamente Marvin impiega ben 2 minuti e 19 secondi in più per eseguire il programma; anche in questo caso quasi tutto il tempo è impiegato per il calcolo effettivo. Il mini-cluster impiega circa 13 minuti per eseguire il programma, e circa 12 minuti è il tempo di calcolo. Si osserva quindi che per problemi ben parallelizzabili e su tempi relativamente lunghi, il mini-cluster offre un certo vantaggio rispetto all'uso di una singola macchina, sebbene più potente, come Marvin.

A questa analisi si aggiunge qualche considerazione sui costi dell'hardware: sia il costo di Marvin, sia il costo del mini-cluster composto da 15 Raspberry, è stato di circa 900€. Chiaramente i costi possono variare a seconda del momento dell'acquisto, del rivenditore, di componenti scelti e di eventuali accessori. Per valutare l'economicità dei diversi apparati, è da non sottovalutare il fatto che ogni Raspberry fornisce una licenza gratuita del software Mathematica $\text{\textcircled{R}}$, che permette di effettuare il calcolo simbolico in modi che sarebbero assai costosi su un Personal Computer tradizionale.

Alla luce di queste considerazioni, la parallelizzazione su un mini-cluster si rivela particolarmente vantaggiosa rispetto all'uso di un solo computer, sebbene quest'ultimo sia molto più potente di ciascun Raspberry e ci si aspetta di poter ulteriormente esplorare maggiormente questo aspetto per risolvere problemi scientifici che richiedono una discreta potenza di calcolo.

$$T_F = 10^5$$

tempi su Raspberry		tempi su Marvin	
real	1m53s	real	1m32s
cpu	1m13s	cpu	1m32s

$$T_F = 10^6$$

tempi su Raspberry		tempi su Marvin	
real	13m9s	real	15m27s
cpu	11m55s	cpu	15m25s

Riferimenti bibliografici

- [1] Benettin, G. e Giorgilli, A.: *On the Hamiltonian interpolation of near-to-the-identity symplectic mappings with application to symplectic integration algorithms*, Journal of Statistical Physics, vol **74** n. 5/6, 1117-1143 (1994).
- [2] Celletti, A.: *Stability and Chaos in Celestial Mechanics*, Springer, 2010.
- [3] Giorgilli, A.: *Metodi e Modelli Matematici per le Applicazioni*, <http://www.mat.unimi.it/users/antonio/metmod/metmod.html>.
- [4] Hénon, M. Heiles, C.: *The applicability of the third integral of motion: Some numerical experiments*, The Astronomical Journal, vol **69**: 73-79 (1964).
- [5] Laskar, J. Robutel, P.: *High order symplectic integrators for perturbed Hamiltonian systems*, Celestial Mechanics and Dynamical Astronomy, vol **80**: 39-62 (2001).
- [6] Locatelli U.: *Appunti sul modello di Hénon e Heiles*, http://www.mat.uniroma2.it/~locatell/master_STS/note_HH_e_int_simpl.pdf.