

# **Stima della costante di Hubble**

*partendo dalle curve di luce di 4 stelle cefeidi  
osservate in 4 diverse galassie*

Valerio Oss

Corso di Laurea in Astronomia

28 gennaio 2018

## Abstract

### STIMA della COSTANTE di HUBBLE

Scopo generale di questa esercitazione è la creazione di un software in Fortran che permetta di stimare la costante di Hubble partendo da una serie di dati forniti.

Avremo a disposizione i dati relativi a una curva di luce di 4 Cefeidi, da cui derivare i relativi periodi di pulsazione e magnitudine media.

Sarà poi necessario calcolare la magnitudine assoluta delle 4 Cefeidi in esame utilizzando i dati di un catalogo (fornito) di Cefeidi, grazie alla relazione periodo-luminosità. A questo punto, calcolare la distanza, utilizzando il modulo di distanza.

Ultimo passo sarà la stima della costante di Hubble (con relativo errore) sfruttando i dati forniti relativi alla velocità di recessione delle 4 galassie a cui queste 4 Cefeidi appartengono.

## Introduzione storica e analisi del sistema fisico

Prima di iniziare ad approfondire il programma in Fortran per il calcolo della costante di Hubble, dobbiamo analizzare il sistema fisico in gioco e introdurre storicamente il problema.

Possiamo innanzitutto dividere la nostra analisi in 4 fasi principali:

1. Analisi della curva di luce di ogni singola Cefeide fornita dal problema, per calcolare la magnitudine apparente media e il periodo di pulsazione.
2. Calcolo della magnitudine assoluta delle 4 Cefeidi partendo dall'analisi dei dati della lista di Cefeidi per il calcolo della relazione periodo-luminosità.
3. Calcolo della distanza della galassia a cui la Cefeide appartiene
4. Calcolo della costante di Hubble  $H_0$ . Va fatto notare che il parametro  $H$  è relativo al tempo  $t$  (ad esempio ad oggi), quindi  $H(t)$  varia in funzione del tempo ed è quindi più corretto in generale parlare di parametro di Hubble  $H(t)$ . Nel nostro caso, il calcolo è effettuato per  $t = \text{oggi}$ , quindi questo valore è da considerarsi costante.

Agli inizi del 1900 l'astronoma Henrietta Leavitt dall'osservatorio di Harvard, osservò alcune stelle appartenenti alla Grande e Piccola Nube di Magellano e stilò un catalogo di 1777 stelle variabili periodiche. Di queste, 47 furono catalogate come Cefeidi: la Leavitt scoprì che questo tipo di stelle variabili, che prendono il nome dalla stella Delta Cephei (della costellazione del Cepheus, leggendario re di Etiopia), hanno una relazione tra il loro periodo e la luminosità. In pratica quelle con periodo più lungo erano più brillanti di quelle con periodo più corto. Erano inoltre nella stessa galassia nana, quindi approssimativamente alla stessa distanza da noi, il che comportava che ogni differenza di magnitudine apparente era legata a una differenza di magnitudine assoluta.

Facendo un grafico delle sue osservazioni la Leavitt scoprì che c'era una relazione lineare fra il logaritmo della luminosità (la magnitudine assoluta  $M_V$ ) e il logaritmo del periodo  $P$ :

$$M_V = c_1 \log(P) + c_2$$

Dove:  $M_V$  è la magnitudine assoluta,  $P$  è il periodo,  $c_1$  e  $c_2$  sono due coefficienti da calcolare tramite regressione lineare

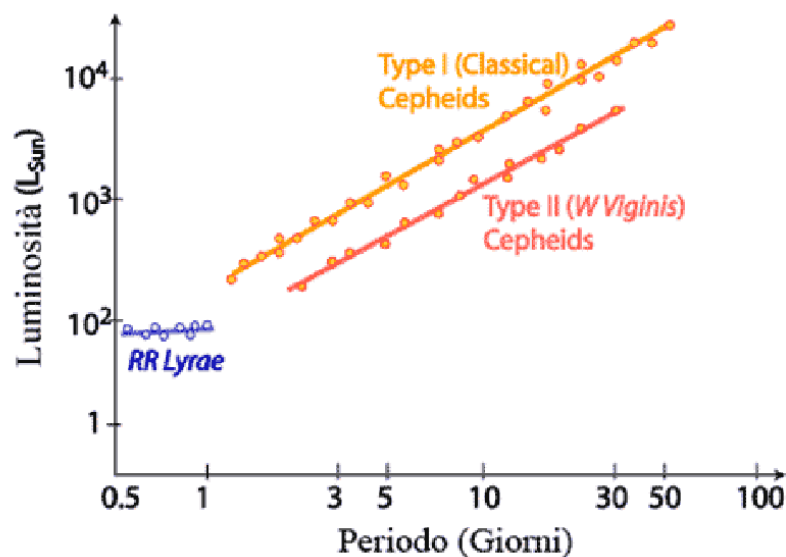


Figura 1: Relazione periodo-luminosità per le variabili Cefeidi di tipo I e II

Questa relazione permette di utilizzare le Cefeidi come indicatore di distanza ed è per questo motivo che sono chiamate *candele standard*, cioè oggetti astronomici con luminosità *conosciuta*.

Una volta nota la magnitudine assoluta della Cefeide tramite la relazione precedente, è possibile calcolarne la distanza tramite il modulo di distanza

$$\mu = m_v - M_v$$

e tramite la relazione tra i flussi possiamo scrivere che tale differenza è

$$\mu = m_v - M_v = 5 \log_{10} d - 5$$

Con  $d$  = distanza dell'oggetto in *parsec* dove il *parsec* è l'unità di lunghezza che rappresenta la distanza dalla Terra di una stella che ha una parallasse annua di 1 secondo d'arco.

E' facile quindi derivare la distanza in parsec come:

$$d = 10^{0.2\mu+5}$$

L'ultima fase riguarda il calcolo della *costante di Hubble*.

Edwin Hubble nel 1929 enunciò la relazione lineare tra lo spostamento verso il rosso della luce emessa dalle galassie e la loro distanza: più sono lontane le galassie e più il loro spettro è spostato verso il rosso (*redshift*). In pratica tutto ci fa capire che l'universo si sta espandendo e che tutte le galassie si stanno allontanando una dall'altra.

La legge di Hubble è espressa dalla relazione matematica:

$$v_{rec} = H_0 d$$

Dove  $v_{rec}$  è la velocità di recessione della galassia,  $d$  la distanza della galassia e  $H_0$  la costante di Hubble (espressa in km/sec/Mpc), che si può ottenere tramite un fit lineare tra le velocità di recessione (fornite) e le distanze calcolate in precedenza.

## Dati disponibili

I dati a nostra disposizione per questo progetto sono:

1. 4 tabelle per ognuna delle Cefeidi, contenenti le curve di luce, cioè l'andamento della magnitudine apparente in banda V (visibile) in funzione del giorno giuliano (Julian Day, JD - numero di giorni passati dal mezzogiorno del 1° Gennaio 4713 a.C.).
2. Una tabella contenente un catalogo di Cefeidi generiche con nome, periodo in giorni, magnitudine assoluta in banda V.
3. Una tabella contenente la velocità di recessione e il relativo errore (in km/s) delle 4 galassie a cui le Cefeidi appartengono.

## Algoritmi numerici utilizzati

Sono stati utilizzati alcuni algoritmi numerici implementati quindi in Fortran per risolvere il problema in esame, e in particolare:

1. Interpolazione numerica con il metodo delle spline cubiche
2. Metodo di Gauss per la risoluzione di sistemi lineari
3. Regressione lineare o fit lineare (pesato e non)

Ciascun algoritmo è stato implementato in Fortran 95 su un sistema operativo Linux (Ubuntu 16.04), con il compilatore gfortran, il compilatore GNU del linguaggio di programmazione omologo. Il programma completo contiene tutti gli algoritmi sotto forma di subroutines chiamate da un main program che si occupa della gestione generale (input/output, definizione variabili).

## 1. Interpolazione numerica con il metodo delle spline cubiche

L'interpolazione consiste nel processo di *stimare* valori intermedi di una funzione, posizionati in mezzo a dati conosciuti in modo preciso. Se abbiamo a disposizione  $(n+1)$  dati, allora esisterà uno ed un solo polinomio di grado  $n$  che passa per tutti i punti. Esistono molteplici metodi di interpolazione, ad esempio tramite i polinomi di Newton o i polinomi di Lagrange. Nel nostro caso però utilizzeremo il metodo di *interpolazione con spline*. Tale metodo si basa sull'applicazione di un polinomio di ordine più basso differente per ogni intervallo tra i punti. L'interpolazione con spline fornisce così un andamento più *smooth* tra i dati. Esistono vari tipi di interpolazione con spline: *lineare* (ogni intervallo è connesso con una *linea retta*), *quadratica* (ogni intervallo è rappresentato da un diverso *polinomio del secondo ordine*) e *cubica*, quella implementata in questo progetto.

Nella spline cubica, ogni intervallo è rappresentato da un diverso *polinomio del terzo ordine*.

$$f(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

Se abbiamo  $(n+1)$  punti, avremo  $n$  intervalli e quindi  $4 \times n$  incognite da trovare.

Ovviamente perché tale algoritmo possa funzionare, deve sottostare ad alcune condizioni, e cioè:

1. Le 2 funzioni adiacenti devono *passare per i punti interni*:  $2(n-1)$  equazioni per  $i=2, n$
2. La prima e l'ultima funzione devono *passare per i punti iniziale e finale*: 2 equazioni
3. Le *derivate prime* nei punti interni devono essere uguali:  $(n-1)$  equazioni per  $i=2, n$
4. Le *derivate seconde* nei punti interni devono essere uguali:  $(n-1)$  equazioni
5. Le *derivate seconde* nei nodi agli estremi devono essere assunte nulle: 2 equazioni (condizioni arbitrarie dette di *spline naturale*)

Alla fine avremo  $4n$  equazioni lineari da risolvere simultaneamente, cioè un *sistema lineare algebrico*, per trovare i  $4n$  coefficienti.

E' importante che i dati a disposizione siano in ordine crescente: non è il caso del nostro set di dati delle Cefeidi, che andranno quindi prima di tutto ordinati con un *algoritmo di sorting*, in particolare utilizzando il metodo di *Bubble Sort*, cioè un algoritmo per confronto tramite scambio di elementi.

Serve quindi costruire la matrice rappresentativa del sistema lineare algebrico, le cui incognite sono le derivate seconde negli  $(n-1)$  punti interni, secondo l'equazione:

$$\begin{aligned} & (x_i - x_{i-1})f''(x_{i-1}) + 2(x_{i+1} - x_{i-1})f''(x_i) + (x_{i+1} - x_i)f''(x_{i+1}) = \\ & = \frac{6}{(x_{i+1} - x_i)} [f(x_{i+1}) - x_i] + \frac{6}{(x_i - x_{i-1})} [f(x_{i-1}) - x_i] \end{aligned}$$

e quindi risolvere il sistema lineare algebrico trovando tutte le  $(n+1)$  derivate seconde dei nodi, per ogni generico punto  $x$  in cui si vuole fare l'interpolazione. Serve trovare prima l'intervallo a cui appartiene  $x$

$$x_{i-1} < x < x_i$$

A questo punto il valore interpolato della funzione in  $x$  si ottiene dalla formula:

$$\begin{aligned} f_i(x) &= \frac{f''(x_{i-1})}{6(x_i - x_{i-1})} (x_i - x)^3 + \frac{f''(x_i)}{6(x_i - x_{i-1})} (x - x_{i-1})^3 + \\ & \left[ \frac{f(x_{i-1})}{(x_i - x_{i-1})} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6} \right] (x_i - x) + \\ & \left[ \frac{f(x_i)}{(x_i - x_{i-1})} - \frac{f''(x_i)(x_i - x_{i-1})}{6} \right] (x - x_{i-1}) \end{aligned}$$

La soluzione del sistema lineare algebrico appena descritto si può ottenere con il metodo di Gauss, che descriviamo nel prossimo paragrafo.

## 2. Sistema lineare algebrico con metodo di Gauss

Un sistema di equazioni lineari può essere descritto tramite una matrice detta matrice completa dei coefficienti del sistema. Per creare un algoritmo che utilizzi il metodo di Gauss per la risoluzione di sistemi lineari algebrici bisogna seguire alcuni step; è necessario infatti trasformare la matrice in esame in una matrice a scalini tramite le *mosse di Gauss*.

*Le mosse di Gauss* sono operazioni che modificano la matrice in uno dei seguenti modi: scambio di due righe, moltiplicazione di una riga per un numero diverso da zero, somma di una riga con un multiplo di un'altra riga.

Si può trasformare una qualunque matrice in una matrice a scalini tramite le seguenti mosse:

1. Se la prima riga ha il primo elemento nullo, si scambia con una riga che ha il primo elemento non nullo. Se tutte le righe hanno il primo elemento nullo, si va al punto 4.
2. Per ogni riga  $A_i$  con primo elemento non nullo, eccetto la prima ( $i > 1$ ), si moltiplica la prima riga per un coefficiente scelto in maniera tale che la somma tra la prima riga e  $A_i$  abbia il primo elemento nullo (quindi coefficiente =  $-A_{i1}/A_{11}$  )
3. Si sostituisce  $A_i$  con la somma appena ricavata.
4. Ora sulla prima colonna dovremmo avere, eccetto forse la prima, valori nulli. A questo punto si ritorna al punto 1 considerando la sottomatrice che si ottiene cancellando la prima riga e la prima colonna.

Avremo così una matrice a scalini che identificherà ancora i coefficienti del nostro sistema lineare ma che sarà molto più facilmente risolvibile e che utilizzeremo per il calcolo delle spline (vedi algoritmo 1).



### 3. Regressione lineare o fit lineare

La regressione lineare rappresenta un metodo di stima del valore atteso. Se due grandezze  $x$  e  $y$  sono legate tra loro da una relazione lineare, del tipo

$$y = ax + b$$

Dove  $a$  e  $b$  sono i coefficienti della retta, che sono appunto incognite da determinare, noti invece un insieme di valori  $x$  e  $y$ . Nel nostro caso abbiamo già a disposizione un catalogo di Cefeidi con relativo periodo e magnitudine assoluta in banda V. Calcolando quindi da questo set di dati il fit lineare, cioè determinando i coefficienti  $a$  e  $b$  della retta che approssima maggiormente i dati disponibili, potremo poi calcolare la magnitudine assoluta delle nostre cefeidi, conoscendone il periodo.

I nostri dati devono avere incertezze della grandezza  $x$  trascurabili e quelle della grandezza  $y$  tutte della stessa dimensione.

La probabilità di ottenere  $y_i$  sarà:

$$P_{a,b}(y_i) \propto \frac{1}{\sigma_y} e^{-(y_i - ax_i - b)^2 / 2\sigma_y^2}$$

Che dipende da  $a$  e  $b$  (incogniti). La probabilità di ottenere il nostro insieme completo di misure  $y_1 \dots y_n$  è il prodotto

$$P_{a,b}(y_1 \dots y_n) = P_{a,b}(y_1) \dots P_{a,b}(y_n)$$

che è proporzionale

$$P_{a,b}(y_1 \dots y_n) \propto \frac{1}{\sigma_y^N} e^{-\chi^2 / 2}$$

dove ora chi-quadro è

$$\chi^2 = \sum_1^N \frac{(y_i - ax_i - b)^2}{\sigma_y^2}$$

Quindi ora perché la  $P$  sia massima,  $\chi$  deve essere minima, quindi derivando rispetto ad  $a$  e  $b$  e ponendo uguale a zero otteniamo un sistema di equazioni simultanee per  $a$  e  $b$  che come risultato daranno

$$a = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{N \sum x_i^2 - (\sum x_i)^2} \quad b = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{N \sum x_i^2 - (\sum x_i)^2}$$

Le incertezze nella misura di  $y$  saranno

$$\sigma_y = \sqrt{\frac{1}{N-2} \sum (y_i - ax_i - b)^2}$$

mentre quelle di  $a$  e  $b$

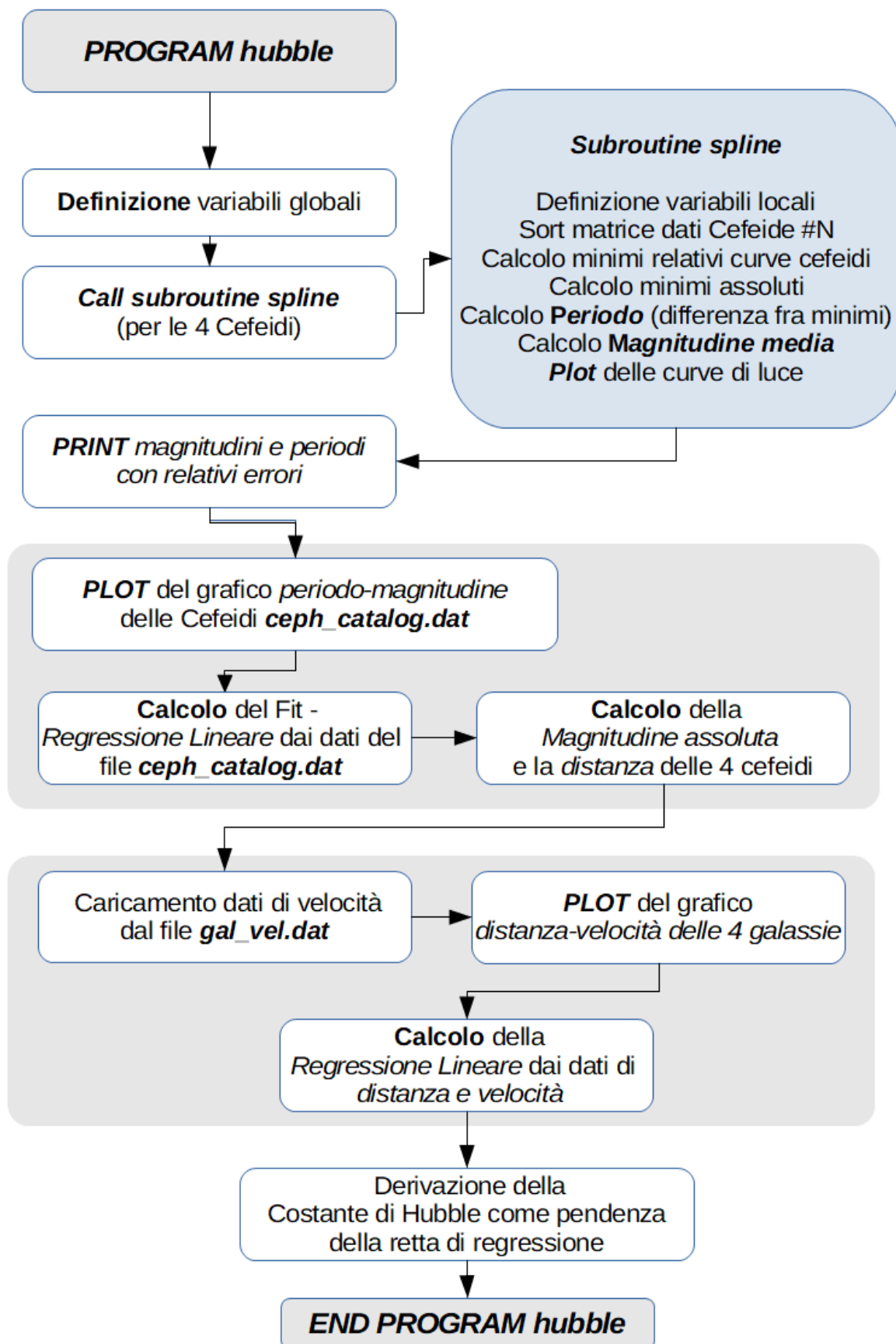
$$\sigma_a = \sigma_y \sqrt{\frac{N}{N \sum x_i^2 - (\sum x_i)^2}} \quad \sigma_b = \sigma_y \sqrt{\frac{\sum x_i^2}{N \sum x_i^2 - (\sum x_i)^2}}$$

La regressione lineare e quindi i valori di  $a$  e  $b$  calcolabili con queste formule saranno utilizzati in due momenti differenti del software:

1. Partendo dal file **ceph\_catalog.dat**, si calcola la pendenza della retta (e intersezione con gli assi) di un set noto di Cefeidi. Tale pendenza servirà per derivare la magnitudine assoluta di ciascuna delle 4 Cefeidi in esame, partendo dal periodo, e quindi, grazie al *modulo di distanza*, della reale distanza in parsec.
2. Partendo dai dati di velocità del file **gal\_vel.dat** e dei valori delle distanze calcolate nel punto precedente si calcola la pendenza della retta (e intersezione con gli assi), che è appunto il valore della *costante di Hubble*. In questo caso però essendo i dati delle velocità con errore (diverso per ogni  $V$ ), dovremo usare il metodo dei minimi quadrati *pesato*.

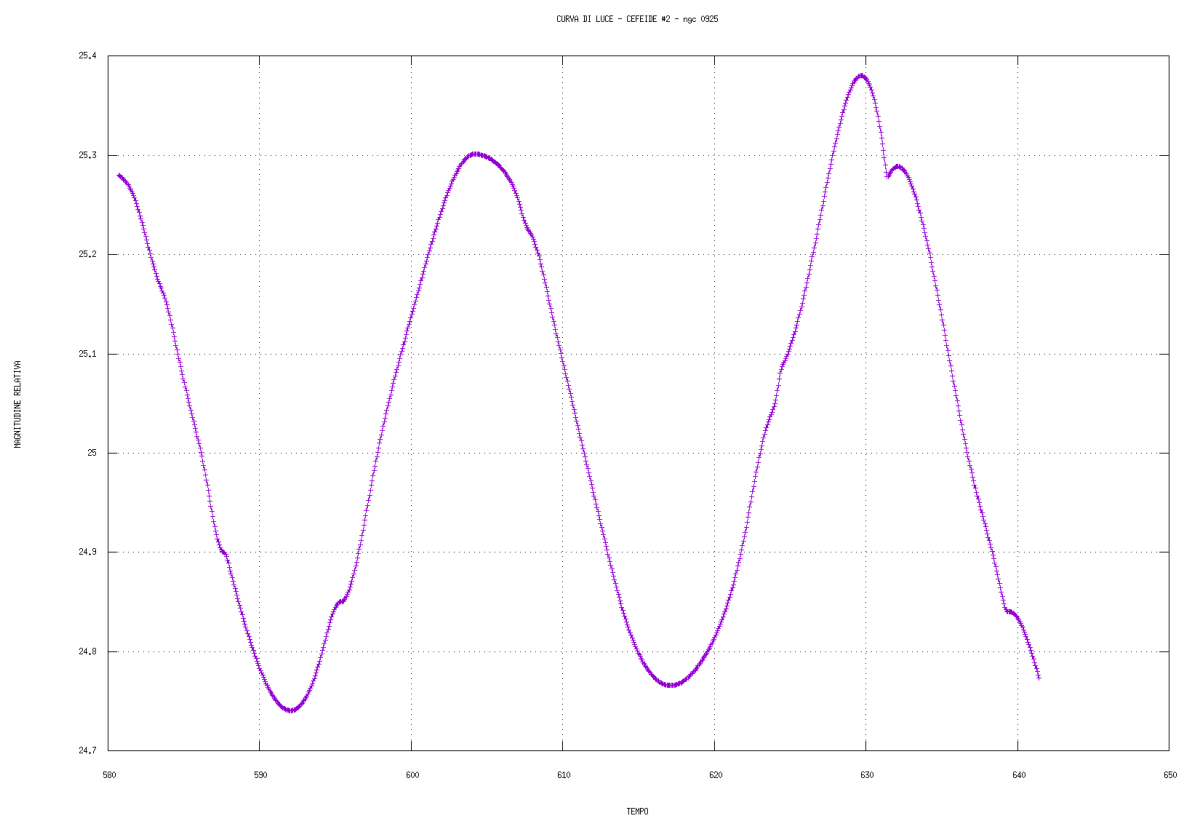
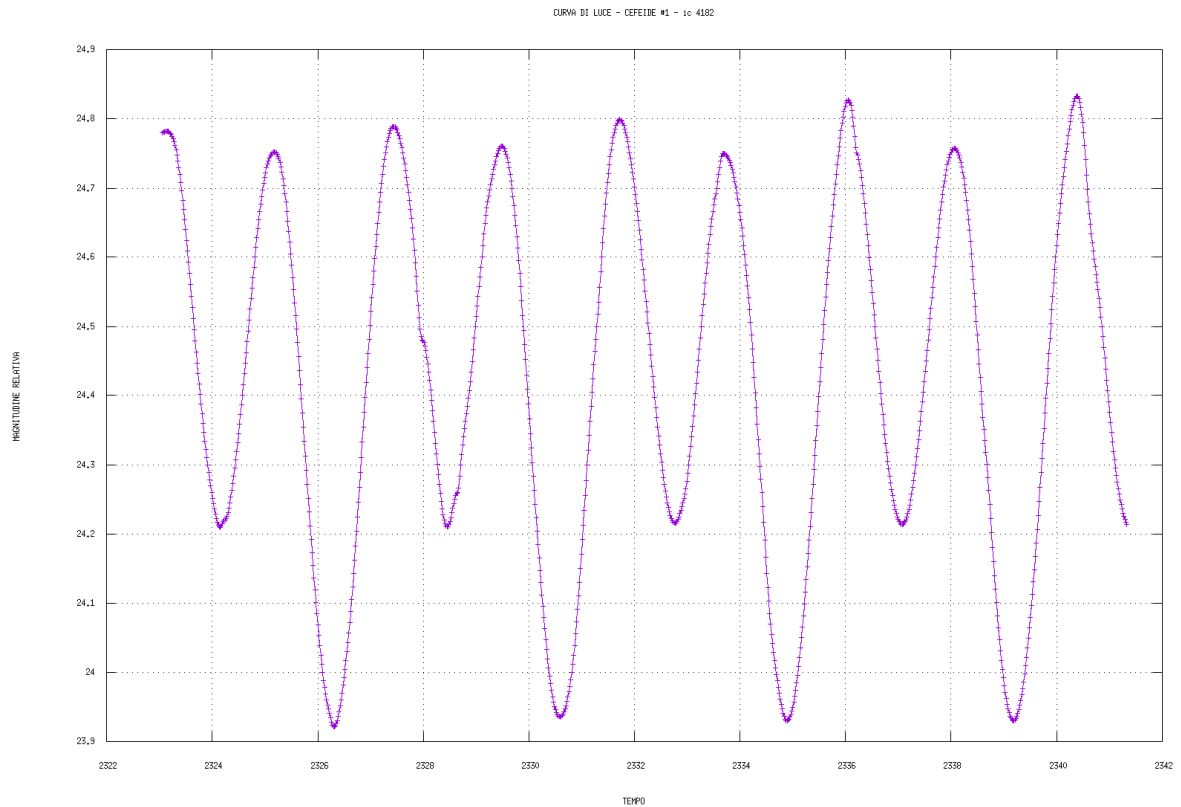
## Analisi software

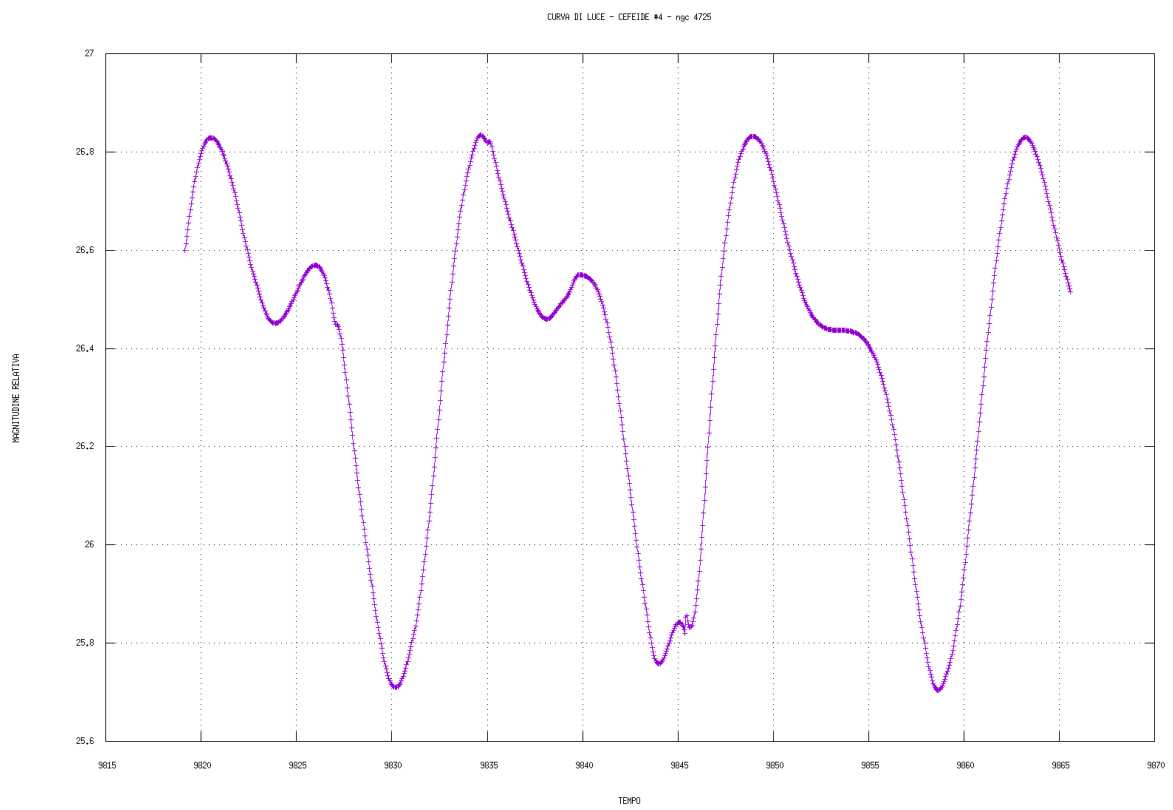
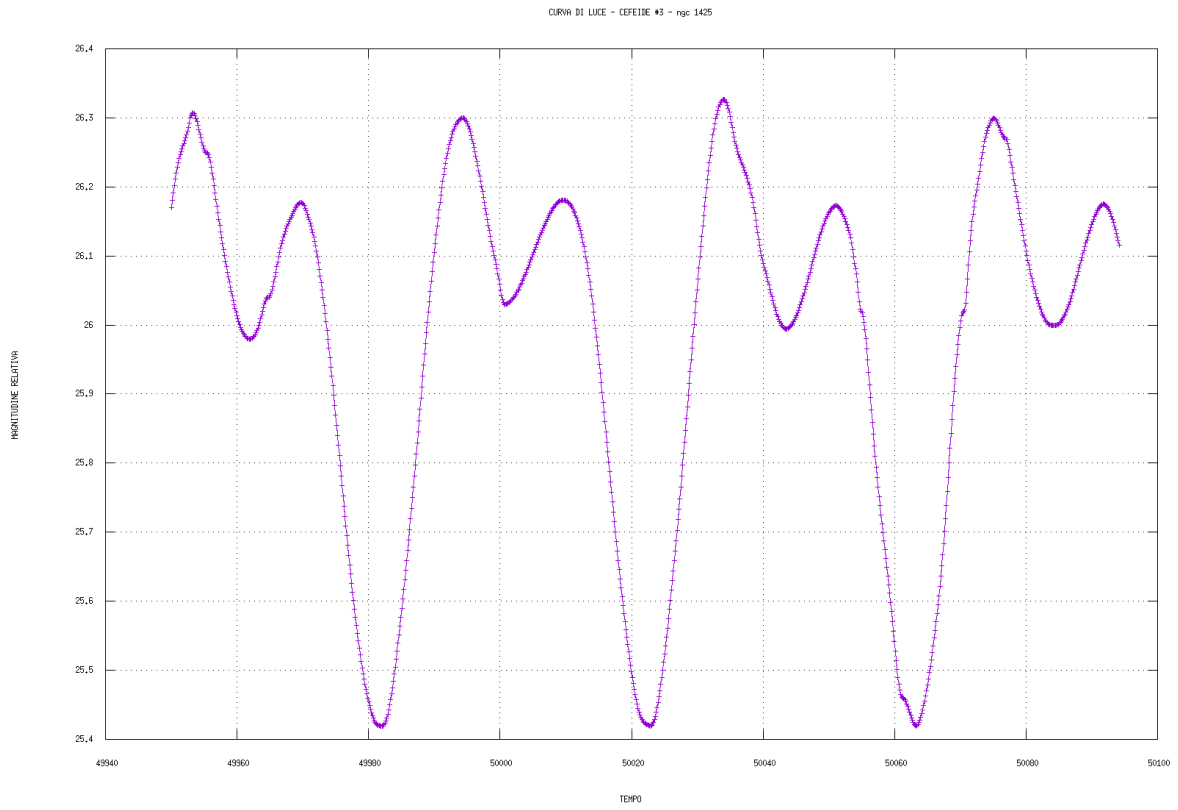
Come già accennato per lo sviluppo del programma *Stima della costante di Hubble* è stato usato il linguaggio di programmazione Fortran 95 su Ubuntu Linux 16.04, con il compilatore gfortran, uno dei più utilizzati in ambito di ricerca e sviluppo scientifico. L'analisi del progetto software è stata fatta considerando i dati a disposizione e i risultati dell'output da calcolare. Ne è risultato il seguente diagramma di flusso:



## Risultati

Grazie ai dati a disposizione relativi alle 4 Cefeidi, si è potuto plottare (con una suddivisione di 1000 steps) le curve di luce delle 4 stelle variabili.



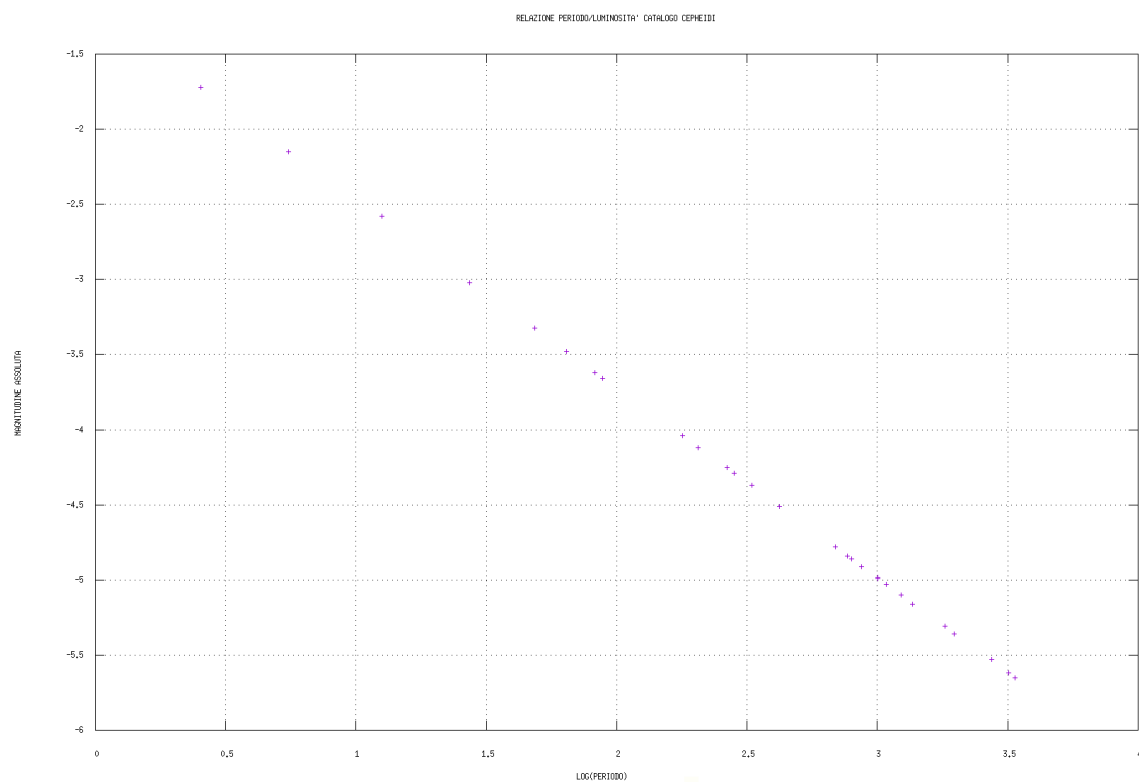


Dall'analisi delle curve di luce, e in particolare dai minimi assoluti delle curve, sono stati calcolati i periodi, come media delle differenze fra i minimi, compresi di errore calcolati come deviazione standard. Il risultato è riportato nella seguente tabella:

Tabella 1  
*Periodi in giorni e Magnitudine apparente media delle Cefeidi con relativo errore*

Nome Galassia/Cefeide	Periodo	Errore Periodo	Mag.apparente media	Errore Mag.apparente
IC 4182	4.2944	$\pm 0.0106$	24.4236	$\pm 0.2381$
NGC 0925	25.2799	$\pm 0.2578$	25.0053	$\pm 0.1671$
NGC 1425	40.5895	$\pm 0.3059$	25.9587	$\pm 0.2321$
NGC 4725	14.2015	$\pm 0.5588$	26.3518	$\pm 0.2871$

Tramite la funzione **gnuplot** di Fortran è possibile plottare il grafico della relazione fra periodo e luminosità del set di dati forniti:



Come si può facilmente notare questo grafico segue un andamento rettilineo: possiamo quindi utilizzare i dati a nostra disposizione per calcolare i valori di luminosità assoluta delle 4 Cefeidi del nostro progetto, partendo dal periodo.

Il calcolo della regressione lineare di questo set di dati dà come parametri della retta:

$$A \text{ (intersezione)} = -1.2056 \quad \text{sigma A (errore)} = \pm 0.0026$$

$$B \text{ (pendenza)} = -2.9001 \quad \text{sigma B (errore)} = \pm 0.0030$$

Le magnitudini assolute delle 4 Cefeidi con errori calcolati tramite il metodo della propagazione degli errori sono

Tabella 2

*Magnitudine assoluta delle Cefeidi con relativo errore*

Nome Galassia/Cefeide	Magnitudine assoluta	Errore Magnitudine Assoluta
IC 4182	-3.041	$\pm 0.008$
NGC 0925	-5.274	$\pm 0.030$
NGC 1425	-5.870	$\pm 0.023$
NGC 4725	-4.547	$\pm 0.114$

Le distanze di ciascuna galassia/Cefeide, compresa di errore sono:

Tabella 3

*Distanze delle Galassie/Cefeidi con relativo errore (in Mega parsec)*

Nome Galassia/Cefeide	Distanza [Mpc]	Errore distanza
IC 4182	3.111	$\pm 0.011$
NGC 0925	11.371	$\pm 0.157$
NGC 1425	23.214	$\pm 0.241$
NGC 4725	15.130	$\pm 0.796$

Il calcolo della costante di Hubble è ora possibile tramite la relazione  $v_{rec} = H_0 d$

Possiamo calcolare indipendentemente le 4 costanti, relative ad ogni galassia, date le velocità di recessione (tabella 4) ottenendo così i risultati di tabella 5.

Tabella 4

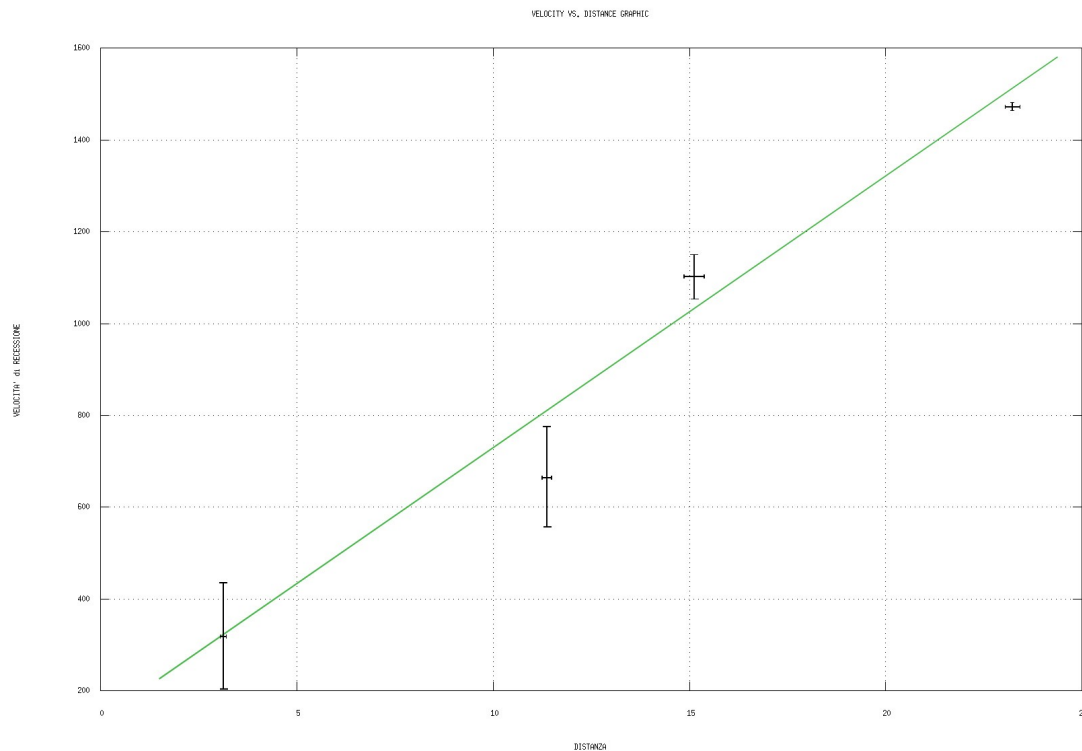
*Velocità di recessione per ciascuna delle Galassie/Cefeidi con relativo errore (fornita)*

Nome Galassia/Cefeide	Velocità di recessione [km/s]	Errore [km/s]
IC 4182	318	$\pm 318$
NGC 0925	664	$\pm 290$
NGC 1425	1473	$\pm 8$
NGC 4725	1103	$\pm 122$

Tabella 5  
*Costante di Hubble per ciascuna delle Galassie/Cefeidi con relativo errore*

Nome Galassia/Cefeide	Costante di Hubble $H_0$ [km/s/Mpc]	Errore
IC 4182	102.210	$\pm 102.210$
NGC 0925	58.394	$\pm 25.516$
NGC 1425	63.453	$\pm 0.742$
NGC 4725	72.900	$\pm 8.928$

Per trovare una stima della costante di Hubble potremmo calcolare la media pesata dei dati della tabella 5 e ottenere quindi un unico valore, pesando di più quelli con errore minore.



E' altresì possibile calcolare la costante utilizzando il metodo dei minimi quadrati pesato sui dati a nostra disposizione, che ci fornisce la pendenza della retta di regressione lineare, che è appunto la nostra  $H_0$  con relativo errore. L'utilizzo del metodo di regressione lineare *pesato* è necessario in quanto i dati a nostra disposizione hanno differenti incertezze per gli  $y_i$  (i dati di velocità di recessione). Otteniamo così la costante di Hubble:

$$H_0 = 54.16 \pm 10.00 \text{ km/s/Mpc}$$



## Conclusioni

Il procedimento di calcolo della costante di Hubble ci ha permesso di verificare una serie di algoritmi numerici e la propagazione degli errori risultanti dai vari passaggi intermedi. La subroutine *spline* per calcolare periodo e magnitudine media delle variabili Cefeidi, ha mantenuto l'errore relativo molto basso (in media dell'ordine dell'1%) anche sul successivo calcolo di Magnitudine assoluta e distanza.

Il grande errore sulle velocità di recessione delle galassie (fornite, nel caso della prima galassia è addirittura del 100%) ha ovviamente incrementato l'errore finale nel calcolo della costante di Hubble, che è quindi risultato (utilizzando il metodo di regressione lineare pesato) dell'ordine del 18%, comunque abbastanza vicino ai valori ufficiali della costante che a tutt'oggi è di  $72 \pm 8$  km/s/Mpc, anche se una recente serie di osservazioni tramite il telescopio Hubble potrebbe aver “raffinato” questo valore a  $74,2 \pm 3,6$  km/s/Mpc, in accordo col precedente ma 2 volte più accurato.