Indice

1 Introduzione

Studiamo la stabilità delle stelle di neutroni in regime relativistico considerando 3 possibili equazioni di stato per la materia. Una volta risolte le equazioni è possibile ottenere l'espressione del potenziale gravitazione della stella e calcolare l'effetto sulla radiazione emessa dalla stessa.

Viene calcolata la radianza per ogni stella a 3 distanze diverse e la potenza totale di emissione in funzione della distanza dalla stella. Viene quindi calcolata la temperatura apparente delle 3 stelle più massive in funzione di quella effettiva e poi viene studiata la temperatura apparente in funzione della pressione centrale della stella.

2 Stabilità

Le equazioni che descrivono la stabilità di una stella in funzione della massa (m) e della pressione (P) sono quelle di Tolman-Oppenheimer-Volkoff

$$\begin{cases} \dfrac{\mathrm{d}P(r)}{\mathrm{d}r} = -G\dfrac{m(r)\epsilon(r)}{r^2c^2}\left(1+\dfrac{P(r)}{\epsilon(r)}\right)\left(+\dfrac{4\pi r^3P(r)}{m(r)c^2}\right)\left(1-\dfrac{2Gm(r)}{rc^2}\right)^{-1} & \text{(1a)} \\ \dfrac{\mathrm{d}m(r)}{\mathrm{d}r} = 4\pi r^2\dfrac{\epsilon(r)}{c^2} & \text{(1b)} \\ \dfrac{\mathrm{d}\Phi(r)}{\mathrm{d}r} = -\dfrac{1}{P(r)+\epsilon(r)}\dfrac{\mathrm{d}P(r)}{\mathrm{d}r} & \text{(1c)} \end{cases}$$

Dove la terza equazione è disaccoppiata e descrive il potenziale gravitazionale della stella. Usiamo 3 diverse densità di energia per la materia della stella (eq. 2b viene presa con due coppie di valori diversi di Γ e K):

$$\epsilon_1(n) = \mu c^2 n + a n_0\left(\frac{n}{n_0}\right)^{\alpha+1} + b n_0\left(\frac{n}{n_0}\right)^{\beta+1} \quad \text{(2a)}$$

$$\epsilon_{2/3}(n) = \mu c^2 n + K c^2 n^{\Gamma} \quad \text{(2b)}$$

$$\text{con} \quad a = 13.4\text{MeV fm}^{-3}\,, \quad \alpha = 0.514, \quad b = 5.62\text{MeV fm}^{-3}\,, \quad \beta = 2.436\,, \quad n_0 = 0.16\text{fm}^{-3} \quad \text{(3)}$$

dove n è la densità numerica, μ la massa del neutrone e quindi $\rho = \mu n$ è la densità di massa.

Visto che la densità di energia è in funzione di n e le incognite del sistema 1 sono P e m dobbiamo scrivere la densità di energia in funzione di queste ultime partendo dalla relazione termodinamica

$$P = -\tfrac{\mathrm{d}E}{\mathrm{d}V} \Rightarrow \begin{cases} P = \alpha a n_0\left(\dfrac{n}{n_0}\right)^{\alpha+1} + \beta b n_0\left(\dfrac{n}{n_0}\right)^{\beta+1} & \text{per } \epsilon_1 \quad \text{(4a)} \\ n = \left(\dfrac{P}{K(\Gamma-1)c^2}\right)^{1/\Gamma} & \text{per } \epsilon_{2/3} \quad \text{(4b)} \end{cases}$$

Nel primo caso (eq. 4a) non è stato possibile invertire l'equazione per trovare n in funzione di P. Facciamo le seguenti sostituzioni per rendere le variabili adimensionali e con valori più vicini a 0.

$$m = M_0\hat{m}, \quad r = R_0\hat{r}, \quad P = P_0\hat{P}, \quad \rho = \rho_0\hat{\rho}, \quad K = \hat{K}\frac{\mu^{\Gamma}}{\rho_0^{\Gamma-1}},$$

$$\begin{cases} \dfrac{\mathrm{d}\hat{P}}{\mathrm{d}\hat{r}} = -\dfrac{(\hat{P}+\hat{\epsilon})(\hat{m}+\hat{r}^3\hat{P})}{\hat{r}^2-2\hat{m}\hat{r}} & \text{(5a)} \\ \dfrac{\mathrm{d}\hat{m}}{\mathrm{d}\hat{r}} = \hat{r}^2\hat{\epsilon} & \text{(5b)} \\ \dfrac{\mathrm{d}\Phi}{\mathrm{d}\hat{r}} = -\dfrac{1}{\hat{P}+\hat{\epsilon}}\dfrac{\mathrm{d}\hat{P}}{\mathrm{d}\hat{r}} & \text{(5c)} \end{cases}$$

Otteniamo il sistema 5, dove le densità di energia diventano

$$\hat{\epsilon}_1(\hat{n}) = \hat{n} + \frac{an_0}{P_0}\,\hat{n}^{\alpha+1} + \frac{bn_0}{P_0}\,\hat{n}^{\beta+1} \qquad \text{dove vale} \quad \hat{P} = \frac{\alpha a n_0}{P_0}\,\hat{n}^{\alpha+1} + \frac{\beta b n_0}{P_0}\,\hat{n}^{\beta+1} \tag{6a}$$

$$\hat{\epsilon}_2(\hat{P}) = \left(\frac{\hat{P}}{\hat{K}(\Gamma-1)}\right)^{1/\Gamma} + \frac{\hat{P}}{\Gamma-1} \tag{6b}$$

Nel caso di ϵ_1 non è possibile invertire la relazione $P(n)$ e quindi si utilizzerà un metodo numerico. Per le politropiche ϵ_2 e ϵ_3 si utilizzano i seguenti valori

	$\hat{K}$	Γ
ϵ_2	5/3	0.05
ϵ_3	2.54	0.01

Infine, i valori delle costanti utilizzate per definire le variabili adimensionali sono i seguenti

$$M_0 = 12.655756 M_\odot \quad R_0 = 20.06145\text{km} \quad \epsilon = P_0 = \rho_0 c^2 = n_0 \mu c^2 = 150.174 \frac{\text{MeV}}{c^2 \text{fm}^3} \tag{7}$$

Per il potenziale gravitazionale Φ si può inoltre trovare una soluzione analitica all'esterno della stella che possiamo mettere in forma adimensionale (eq. 8), dove $\hat{M}$ e $\hat{R}$ sono rispettivamente la massa totale e il raggio della stella in forma adimensionale.

$$\Phi_{\text{ext}}(r) = \frac{1}{2}\log\left(1 - \frac{2GM}{rc^2}\right) \implies \Phi_{\text{ext}}(\hat{r}) = \frac{1}{2}\log\left(1 - \frac{2\hat{M}}{\hat{r}}\right) \qquad \hat{r} \geq \hat{R} \tag{8}$$

3 Curva massa raggio

Cominciamo con il risolvere le prime due equazioni 5a e 5b del sistema adimensionale con il metodo `RK4` (appendice A). Per il caso con equzione di stato più complessa (eq. 6a) viene risolta numericamente l'equazione $\hat{P}(\hat{n})$ a fianco per trovare $\hat{n}$ da un $\hat{P}$ fissato (Appendice B).

Segliamo massa iniziale 0 e pressioni iniziali differenti in modo da trovare soluzioni con R compreso tra i 3 e i 40 km. Il grafico massa raggio trovato viene riportato in figura 1.

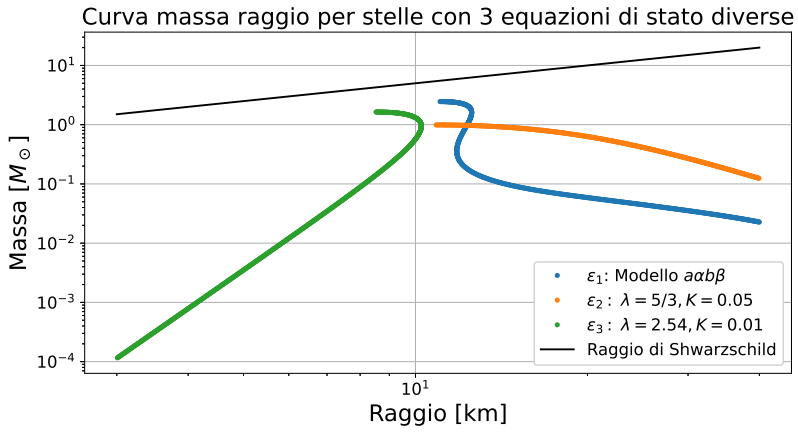

Figura 1: Curva massa raggio per stelle di equzioni di stato $\epsilon_{1/2/3}$. La prima equazione di stato, quella più realistica, prevede stelle di neutroni più massive.

Durante l'esecuzione del programma abbiamo inoltre smesso di incrementare la pressione centrale iniziale P_c quando la condizione di stabilità $\frac{\mathrm{d}M}{\mathrm{d}P_c} > 0$ veniva meno.

Notiamo subito che con l'equazione di stato 2a il modello prevede stelle con massa e raggio maggiore rispetto al limite previsto dagli altri modelli.

	P_0 $[\frac{\text{MeV}}{c^2\text{fm}^3}]$	R [km]	M $[M_\odot]$
ϵ_1	885.2114	11.04289	2.456841
ϵ_2	217.2936	10.86752	0.990100
ϵ_3	948.5211	8.531525	1.635845

Tabella 1: Valori della pressione iniziale P_0, massa totale M e raggio R della stella più massiva per ogni equazione di stato utilizzata.

I valori della stella più massiva per ognuna delle 3 equazioni di stato vengono riportati nella tabella 1.

Ci riferiremo alla stella più massiva ottenuta con la politropica 1 (politropica realistica) come *stella 1* che ha mass M_1 raggio R_1, alla stella con la politropica 2 *stella 2* e così via.

4 Potenziale gravitazionale

Come mostrato nel sistema 1, l'equzione che determina il potenziale gravitazionale è disaccoppiata dalle altre due e, per $r \geq R$, può anche essere risolta analiticamente. Utilizzando le equazioni 5c e 8 possiamo trovare l'espressione generale per il potenziale all'interno della stella riportata in eq. 9.

$$\Phi_{\text{int}}(\hat{r}) = \Phi_{\text{ext}}(\hat{R}) - \int_{\hat{r}}^{\hat{R}} \frac{\mathrm{d}\Phi}{\mathrm{d}x}\,\mathrm{d}x = \frac{1}{2}\log\left(1 - \frac{2\hat{M}}{\hat{R}}\right) + \int_{\hat{r}}^{\hat{R}} \frac{1}{\hat{P}(x) + \hat{\epsilon}(x)} \frac{\mathrm{d}\hat{P}(x)}{\mathrm{d}x}\,\mathrm{d}x \tag{9}$$

dove siamo stati attenti a rendere $\Phi(r)$ continuo per ogni $r \geq 0$ usando il valore di Φ_{ext} in $\hat{R}$. Infine sostituendo alla derivata della pressione l'espressione in 5a otteniamo

$$\Phi_{\text{int}}(\hat{r}) = \frac{1}{2}\log\left(1 - \frac{2\hat{M}}{\hat{R}}\right) + \int_{\hat{r}}^{\hat{R}} \frac{\hat{m}(x) + x^3\hat{P}(x)}{2\hat{m}(x)x - x^2}\,\mathrm{d}x \tag{10}$$

Dati i valori di $\hat{P}(\hat{r})$ e $\hat{m}(\hat{r})$ che si ottengono risolvendo le equazioni di stabilità possiamo quindi risolvere l'integrale con il metodo dei trapezi per ottenere il valore di Φ per ogni r (Appendice C). Il grafico del potenziale gravitazionale ottenuto è mostrato in figura 2.

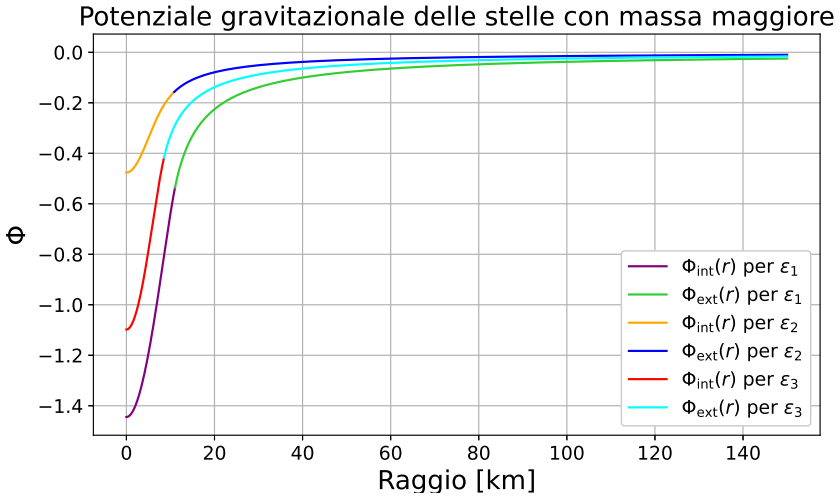

Figura 2: Grafico del potenziale gravitazionale all'interno e all'esterno della stella. Per $r < R$ è stato ottenuto integrando con il metodo dei trapezi l'eq. 10, per $r \geq R$ è stata plottata l'eq. 8

Vediamo dalla figura 2 che in tutti e 3 i casi $\Phi(r)$ è continuo, grazie alla condizione imposta.

Il potenziale Φ (che fa parte del termine $e^{\Phi(r)}c^2\mathrm{d}t^2$ della metrica $\mathrm{d}s^2$, che descrive la geometria dello spazio vicino alla stella) assume un andamento famigliare: raggiunge valori più bassi per le stelle più massive al diminuire di r e tende a 0 per r grandi.

5 Radianza

In metrica di Schwarzschild un fotone emesso a distanza r con frequenza ν_{em} viene ricevuto da un osservatore a distanza r' con una frequeza ν_{ric} data da

$$\frac{\nu_{\text{ric}}}{\nu_{\text{em}}} = e^{\Phi(r)-\Phi(r')}\,. \tag{11}$$

Dove $\Phi(r)$ è propio il potenziale gravitazionale mostrato in figura 2.
La radianza di una stella si può esprimere con l'equazione di Plank per il corpo nero

$$B(\nu, T) = \frac{2h\nu^3}{c^2}\frac{1}{e^{h\nu/(k_B T)} - 1} \tag{12}$$

dove T è la temperatura della stella e h la costante di Plank. Per fare i conti considerando solo il caso con $K_B T = 1\text{MeV}$ e mettiamo l'equzione 12 in funzione di variabili adimensionali. Otteniamo

$$\hat{B}(\hat{\nu}) = \frac{\hat{\nu}^3}{e^{\hat{\nu}} - 1} \tag{13}$$

dove abbiamo definito

$$\nu_0 = \frac{\nu}{\hat{\nu}} = \frac{1\text{MeV}}{h} \simeq 2.417\,989 \times 10^{20}\text{Hz} \tag{14}$$

$$B_0 = \frac{B}{\hat{B}} = \frac{2h\nu_0^3}{c^2} = \frac{2\text{MeV}}{(hc)^2} \simeq 1.301\,059 \times 10^{-6}\frac{\text{MeV}}{\text{fm}^2} \tag{15}$$

Infine, se consideriamo l'effetto doppler dovuto al potenziale gravitazionale descritto in eq. 11 e utilizziamo la formula analitica per il potenziale gravitazionale all'esterno della stella (eq. 8), otteniamo

$$\hat{\nu}_{\text{em}} = \hat{\nu}_{\text{ric}}\left(\frac{1-\frac{2\hat{M}}{r}}{1-\frac{2\hat{M}}{\hat{R}}}\right)^{1/2} \quad\Longrightarrow\quad \hat{B}(\hat{\nu}_{\text{ric}}, \hat{r}) = \frac{\hat{\nu}_{\text{ric}}^3\left(1-\frac{2\hat{M}}{\hat{r}}\right)^{3/2}\left(1-\frac{2\hat{M}}{\hat{R}}\right)^{-3/2}}{\exp\left(\hat{\nu}_{\text{ric}}\left(1-\frac{2\hat{M}}{\hat{r}}\right)^{1/2}\left(1-\frac{2\hat{M}}{\hat{R}}\right)^{-1/2}\right) - 1} \tag{16}$$

Nel figure 3, 4 e 5 viene studiato lo spettro della radiazione emessa per le 3 stelle massive di cui abbiamo studiato il potenziale in 2 a distanze diverse dalla stella. In blu è plottata l'eq. 12, ovvero la radianza senza correzioni relativistiche.

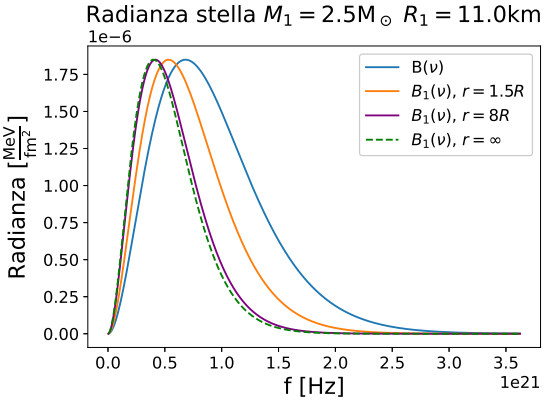

Figura 3: Radianza di una stella con $M = 2.5M_\odot$ e $R = 11.0$km, percepita a 16.5km, 88km e a distanza infinita. Essendo la stella con massa maggiore è anche quella in cui la curva è più spostata verso sinistra.

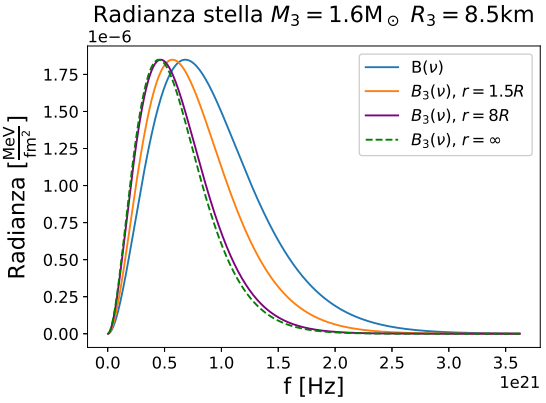

Figura 4: Radianza da una stella con $M = 1.6M_\odot$ e $R = 8.5$km, percepita a 12.8km, 68km e a distanza infinita. In blu la radianza senza correzioni relativistiche.

La curva in tutti e 3 i casi subisce uno spostamento verso le frequenze più basse, *redshift* per l'appunto, e l'effetto è tanto maggiore quanto più la stella è massiva e l'osservatore distante. Già a

una distanza di 8 volte il raggio della stella la curva della radianza è molto simile a quella all'inifinito, in cui il termine $\Phi(r')$ in eq. 11 è trascurabile.

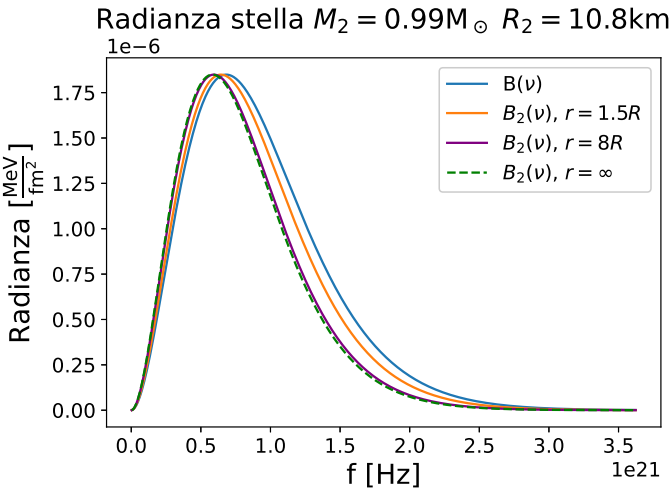

Figura 5: Radianza da una stella con $M = 0.99M_\odot$ e $R = 10.8$km, percepita a 16.2, 86.4 km e a distanza infinita. In blu la radianza senza correzioni relativistiche.

6 Potenza emessa

La potenza totale emessa dalla stella per un elemento della sua superficie, si può ottenere integrando la radianza $B(\nu, T)$ sulle frequenze e sull'angolo solido, con l'accortezza di moltiplicare per un fattore $\cos(\theta)$ (Legge di Lambert) per proiettare la radiazione lungo la normale alla superficie sottesa da $\mathrm{d}\Omega$.

$$\mathcal{P} = \int_0^\infty \mathrm{d}\nu \int B(\nu, T)\cos(\theta)\ \mathrm{d}\Omega = \int_0^{Ak_BT} B(\nu, T)\ \mathrm{d}\nu \int_0^{2\pi} \mathrm{d}\phi \int_0^1 \mathrm{d}(\cos(\theta))\ \cos(\theta) \tag{17}$$

$$\simeq \pi \int_{\nu_{\mathrm{ric}}=0}^{\nu_{\mathrm{ric}}=Ak_BT} B(\nu_{\mathrm{em}}, T)\ \mathrm{d}\nu_{\mathrm{ric}} \tag{18}$$

Nell'ultimo passaggio (eq. 18) è esplicitato che l'integrazione viene fatta su ν_{ric}, ovvero le frequenze misurate a distanza r, abbiamo messo un upperbound finito all'integrale e, per semplificare i calcoli per il prossimo punto, abbiamo lasciato la dipendenza da ν_{em} in B.

Passiamo quindi alle unità adimensionali, riportate in eq. 20, e facciamo un cambio di variabile all'interno dell'integrale utilizzando la formula del redshift $\nu_{\mathrm{ric}} = \nu_{\mathrm{em}} e^{\Phi(R)-\Phi(r)}$:

$$\hat{\mathcal{P}}(\hat{r}) = \int_{\hat{\nu}_{\mathrm{ric}}=0}^{\hat{\nu}_{\mathrm{ric}}=\frac{AT_0}{\nu_0}} \hat{B}(\hat{\nu}_{\mathrm{em}}, \hat{T})\ \mathrm{d}\hat{\nu}_{\mathrm{ric}} = e^{\Phi(\hat{R})-\Phi(\hat{r})} \int_{\hat{\nu}_{\mathrm{em}}=0}^{\hat{\nu}_{\mathrm{em}}=\frac{AT_0\hat{T}}{\nu_0} e^{\Phi(\hat{R})-\Phi(\hat{r})}} \frac{\hat{\nu}_{em}^3}{\exp\left(\hat{\nu}_{\mathrm{em}}/\hat{T}\right) - 1}\ \mathrm{d}\hat{\nu}_{\mathrm{em}} \tag{19a}$$

$$= \left(1 - \frac{2\hat{M}}{\hat{R}}\right)^{1/2} \left(1 - \frac{2\hat{M}}{\hat{r}}\right)^{-1/2} \int_0^{\frac{AT_0\hat{T}}{\nu_0} e^{\Phi(\hat{R})-\Phi(\hat{r})}} \frac{\nu^3}{e^{\nu/\hat{T}} - 1}\ \mathrm{d}\nu \tag{19b}$$

$$= \left(1 - \frac{2\hat{M}}{\hat{R}}\right)^{1/2} \left(1 - \frac{2\hat{M}}{\hat{r}}\right)^{-1/2} \int_0^{\frac{AT_0\hat{T}}{\nu_0}} \frac{\nu^3}{e^{\nu/\hat{T}} - 1}\ \mathrm{d}\nu \tag{19c}$$

$$= \left(1 - \frac{2\hat{M}}{\hat{R}}\right)^{1/2} \left(1 - \frac{2\hat{M}}{\hat{r}}\right)^{-1/2} \hat{T}^4 \int_0^{\frac{AT_0}{\nu_0}} \frac{\nu^3}{e^{\nu} - 1}\ \mathrm{d}\nu\,. \tag{19d}$$

Dove sono state usate le definizioni di $B(\hat{\nu}, \hat{T})$ (eq. 13) e quella di $\Phi(r)$ (eq. 8).
Nel passaggio 19c si è utilizzato il fatto che nell'integrale $A \to \infty$ e che $e^{\Phi(R)-\Phi(r)} < 1$, ignorare il termine migliora quindi la nostra approssimazione.
Infine del passaggio 19d abbiamo fatto un cambio di variabile $\nu/\hat{T} \to \nu$ per eliminare la dipendenza dalla temperatura nell'intergrale.

$$\mathcal{P}_0 = \frac{\mathcal{P}}{\hat{\mathcal{P}}} = \pi B_0 \nu_0 \simeq 9.883\,290 \times 10^{14}\, \frac{\text{MeV}}{\text{s}\,\text{fm}^2}\,, \qquad T_0 = T/\hat{T} = \frac{\text{MeV}}{k_B}\,. \tag{20}$$

6.1 Convergenza dell'integrale

Dal momento che numericamente non è possibile integrare sulle frequenze fino a $\nu = +\infty$, nei passaggi in eq. 18 e poi in eq. 19 abbiamo utilizzato un parametro A che deve essere sufficientemente grande da poter approssimare in modo ragionevole l'integrale. Riportiamo in eq. 21 l'integrale da calcolare.

$$\mathcal{I} = \int_0^{\frac{AT_0}{\nu_0}} \frac{\nu^3}{e^\nu - 1}\, \mathrm{d}\nu \tag{21}$$

Facendo riferimento alla figura 4 dove in blu è plottata la funzione che dobbiamo integrare possiamo vedere che per $\nu > 3.5 \times 10^{21}$Hz la radianza è quasi nulla. In quell'occasione era stata calcolata $\hat{B}(\hat{\nu})$ con $\hat{\nu}$ tra 0 e 15. Come stima iniziale possiamo quindi integrare fino a $\hat{\nu} = 20$ che corrisponde a

$$\frac{A\,T_0}{\nu_0} = 20 \quad \Longleftrightarrow \quad A = 20\,\frac{\nu_0}{T_0} \simeq 4.835\,978 \times 10^{21}\,\text{MeV}^{-1}\,\text{s}^{-1} \tag{22}$$

Possiamo ottenere un valore migliore valutando la convergenza dell'integrale.

Per prima cosa verifichiamo per quale `N` (numero di step nell'integrazione) il valore di $\mathcal{I}$ converge (il codice si trova in Appendice D). Otteniamo il grafico in figura 6. I valori sono degli scarti rispetto a $\mathcal{I}_{N=1e8}$ (l'integrale calcolato con $N = 1 \times 10^8$) e sono normalizzati rispetto allo stesso. Al contriario di quello che succede di solito, il metodo Simpson (che ha un costo computazionale maggiore), non presenta vantaggi rispetto a quello dei trapezi. Dal codice (Appendice D) che genera i dati mostrati in figura 6 otteniamo un errore relativo minore di 1×10^{-7} per `N_trap = 196` e `N_simp = 262`.

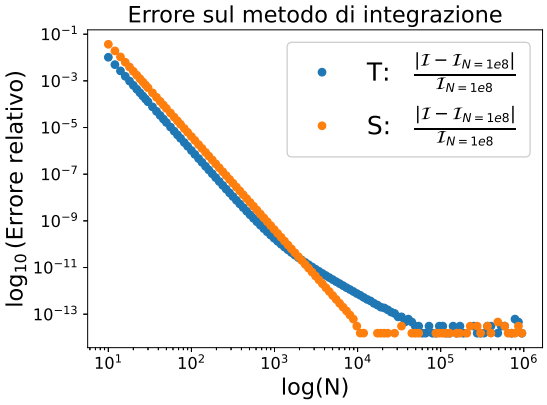

Figura 6: Errore relativo sul calcolo di $\mathcal{I}(1)$ per diversi `N` (numero di step nell'integrazione con i trapezi, **T**, e con simpson, **S**).

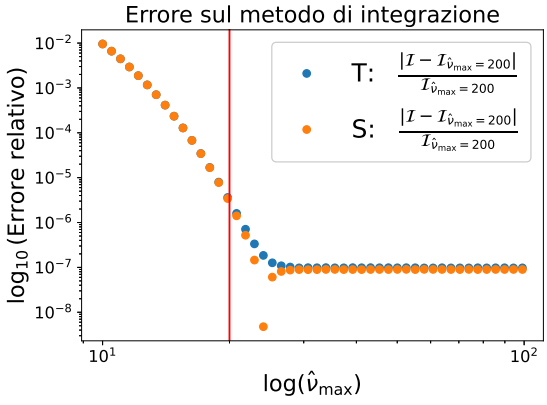

Figura 7: Errore relativo sul calcolo di $\mathcal{I}(1)$ per diversi valori di $\hat{\nu}_{\max}$. Poco dopo la stima iniziale di $\hat{\nu}_{\max} = 20$ (line verticale rossa) l'errore si stabilizza. L'errore non diminuisce come prima perché c'è una sorta di offset dato dall'aver fissato il rapporto `N / nu_max`.

Fissati `N_trap` e `N_simp` per un intervallo $\Delta\hat{\nu} = 20$ proponiamo quindi in figura 7 lo stesso tipo di grafico degli scarti fatto però in funzione della scelta di $\nu_{\max}$, dove abbiamo tenuto costante il valore `N / nu_max` apena ottenuto.

Dallo script per generare i dati mostrati in figura 7 otteniamo un errore di 1×10^{-7} per

Visto che in questo caso il metodo dei trapezi risulta più efficiente possiamo calcolare A rispetto a quest'ultimo

	N	$\hat{\nu}_{\max}$
Trapezi	264	29.2526072
Simpson	312	24.0661923

$$A = 29.2526072\,\nu_0\,\mathrm{MeV}^{-1} \simeq 7.073\,248 \times 10^{21}\,\mathrm{MeV}^{-1}\,\mathrm{s}^{-1} \tag{23}$$

Con questi parametri otteniamo che $\mathcal{I}$ vale:

$$\mathcal{I} = 6.4939388 \tag{24}$$

L'integrale calcolato con Simpson ha lo stesso valore visto l'errore di 1×10^{-7} scelto.

6.2 Potenza totale in funzione della distanza

Studiato l'integrale che serve per il calcolo della potenza possiamo riscrivere $\hat{\mathcal{P}}$ così

$$\hat{\mathcal{P}}(\hat{r}, \hat{T}) = \left(1 - \frac{2\hat{M}}{\hat{R}}\right)^{1/2} \left(1 - \frac{2\hat{M}}{\hat{r}}\right)^{-1/2} \mathcal{I}\,\hat{T}^4 \tag{25}$$

Riportiamo quindi in figura 8 il l'andamento di $\mathcal{P}(r, \hat{T} = 1)$ per le stelle più massive descritte in 1.

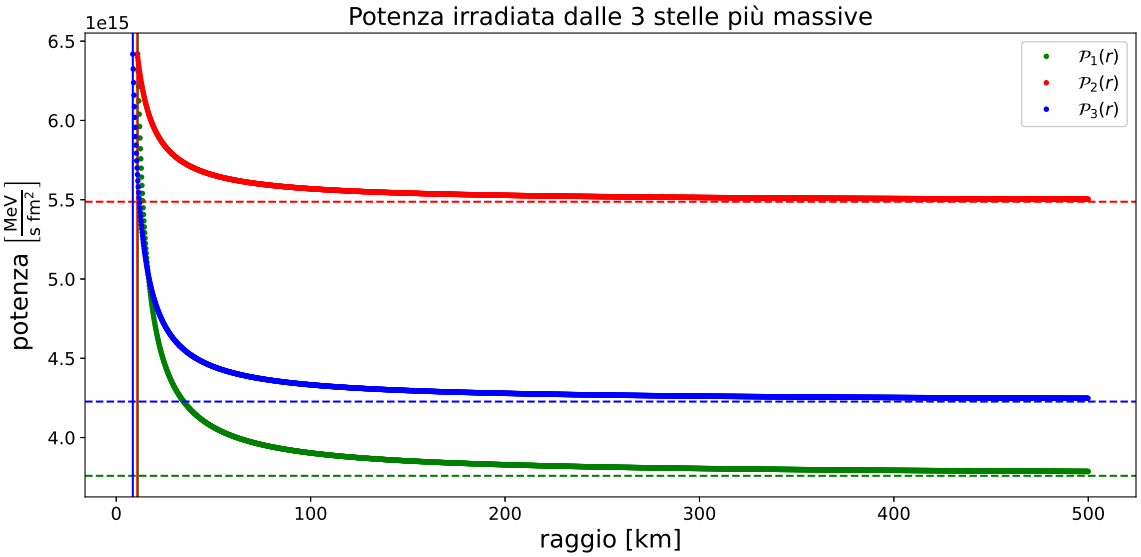

Figura 8: Potenza in funzione della distanza per le 3 stelle: ($R_1 = 59.0$km, $M_1 = 14.3 M_\odot$), ($R_2 = 10.9$km, $M_2 = 0.926 M_\odot$) e ($R_3 = 8.56$km, $M_3 = 1.53 M_\odot$). La linea continua verticale rappresenta il raggio della stella, la linea tratteggiata orizzontate il valore di $\mathcal{P}$ a $r = \infty$. Si noti che la potenza totale emessa è estremamente alta, infatti $1 \times 10^{15}\mathrm{MeV\,s^{-1}\,fm^{-2}} \sim 1 \times 10^{32}\mathrm{W\,m^{-2}}$, ma questo perché abbiamo considerato una temperatura di $k_B T = 1$MeV che corrisponde a $\sim 1 \times 10^{10}$K.

Con linea tratteggiata e linea continua sono rispettivamente rappresentati raggio e valore della potenza all'infinito per ogni stella analizzata.

Vale infine la pena di notare che se consideriamo l'equzione 25 e trascuriamo le correzioni relativistiche otteniamo la legge di Stefan–Boltzmann. Infatti rimettendo le variabili dimensionali

$$\mathcal{P}(T) = P_0 \mathcal{I} \left(\frac{T}{T_0}\right)^4 = \frac{P_0 \mathcal{I} k_B}{\mathrm{MeV}}\,T^4 \qquad \sigma = \frac{P_0 \mathcal{I} k_B^4}{(\mathrm{MeV})^4} = 5.670374\,\mathrm{J\,s^{-1}\,m^{-2}\,K^{-4}} \tag{26}$$

e convertendo P_0 e T_0 in unità del sistema internazionale otteniamo il valore tabulato di σ per le prime 6 cifre decimali (precisione scelta calcolando $\mathcal{I}$).

7 Temperatura Percepita

Una volta ottenuta la potenza totale irradiata da una stella è possibile calcolare la sua temperatura, grazie alla relazione

$$k_B T_{eff} = \left(\mathcal{P}(T) \frac{15 c^2 h^3}{2\pi^5} \right)^{1/4} \tag{27}$$

Dove T_{eff} è la temperatura percepita a una certa distanza dalla stella e T la temperatura reale, che si misurerebbe a $r = R$.

Mettendo le variabili adimensionali, sostituendo a $\hat{\mathcal{P}}$ l'espressione in 25 e ricordando le espressioni di $\hat{\mathcal{P}}_0$ (eq. 20), B_0 (eq. 14) ν_0 (eq. 15) e che $T_0 = \text{MeV}$, otteniamo

$$T_0 \hat{T}_{eff} = \left(\mathcal{P}_0 \frac{15 c^2 h^3}{2\pi^5} \right)^{1/4} \hat{\mathcal{P}}^{1/4} = \frac{15^{1/4}}{\pi} \text{MeV} \left(1 - \frac{2\hat{M}}{\hat{R}} \right)^{1/8} \left(1 - \frac{2\hat{M}}{\hat{r}} \right)^{-1/8} (\hat{T}^4 \, \mathcal{I})^{1/4} \tag{28}$$

$$\hat{T}_{eff}(\hat{r}, \hat{T}) = \frac{15^{1/4}}{\pi} \left(1 - \frac{2\hat{M}}{\hat{R}} \right)^{1/8} \left(1 - \frac{2\hat{M}}{\hat{r}} \right)^{-1/8} \mathcal{I}^{1/4} \, \hat{T} \tag{29}$$

$$\tag{30}$$

Possiamo quindi studiare l'equzione 30 per le 3 stelle studiate in precedenza, questa volta però fissiamo $r = \infty$ e studiamo la T_{eff} in funzione di T, la temperatura propria della stella. Facendo il limite

$$\hat{T}_{eff}(\hat{r}, \hat{T}) = \frac{15^{1/4}}{\pi} \left(1 - \frac{2\hat{M}}{\hat{R}} \right)^{1/8} \mathcal{I}^{1/4} \, \hat{T} \tag{31}$$

Presentiamo il grafico in figura 9, il codice si trova nell'appendice E.

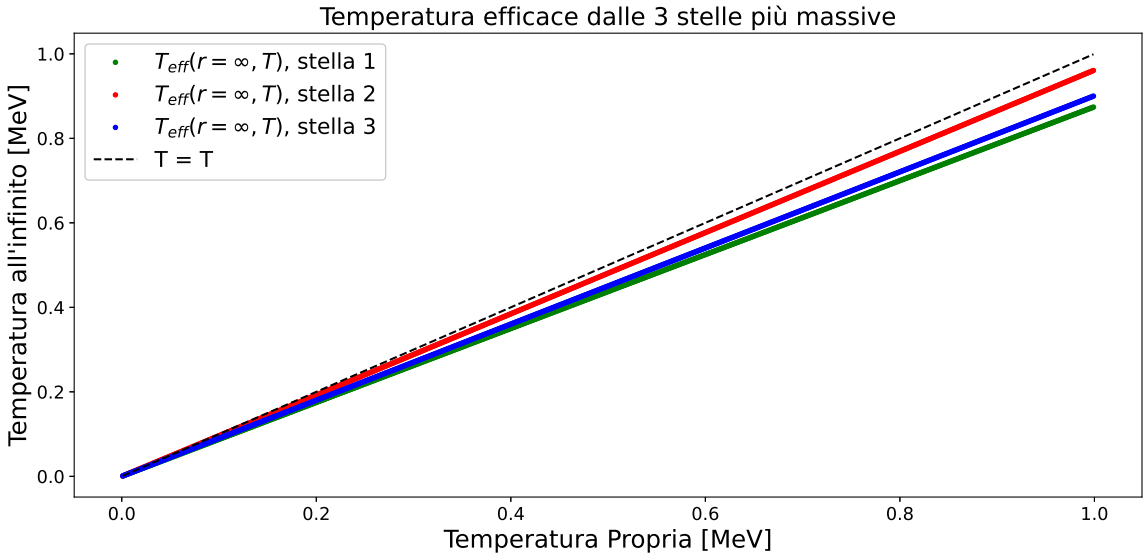

Figura 9: Temperatura percepita all'infinito delle 3 stelle con masse $M_1 = 2.46 M_\odot$, $M_2 = 0.99 M_\odot$ e $M_3 = 1.64 M_\odot$. Come ci aspettavamo la temperatura misurata all'infinito è sempre minore di quella propria della stella.

Come ci aspettavamo la temperatura percepita all'infinito è sempre minore d quella propria della stella. Il grafico mostra anche un'incidenza diversa del redshift per le 3 stelle. In accordo con quello osservato in figura 2 e 8 i fotoni subiscono il redshift maggiore per la stella più massiva $M_1 = 2.46 M_\odot$ e seguono in ordine (sempre di massa) effetti minori per $M_3 = 1.64 M_\odot$ e $M_2 = 0.995 M_\odot$.

8 Temperatura efficace in funzione della pressione centrale

Si presentano infine i grafici della temperatura efficacie in funzione della pressione centrale della stella. Per farlo possiamo usare sempre l'equazione 31 e utilizzare i dati del grafico MR in figura 1. Vengono mostrati i grafici per tutte e 3 le politropiche. Viene inoltre mostrato il caso per la temperatura della stella $\hat{T} = 1$ a sinistra e quello per $\hat{T} = 0.01$ a destra. Come ci si poteva aspettare una pressione centrale maggiore porta a una stella più massiva, il cui redshift gravitazionale è più forte e quindi la temperatura efficacie più bassa.

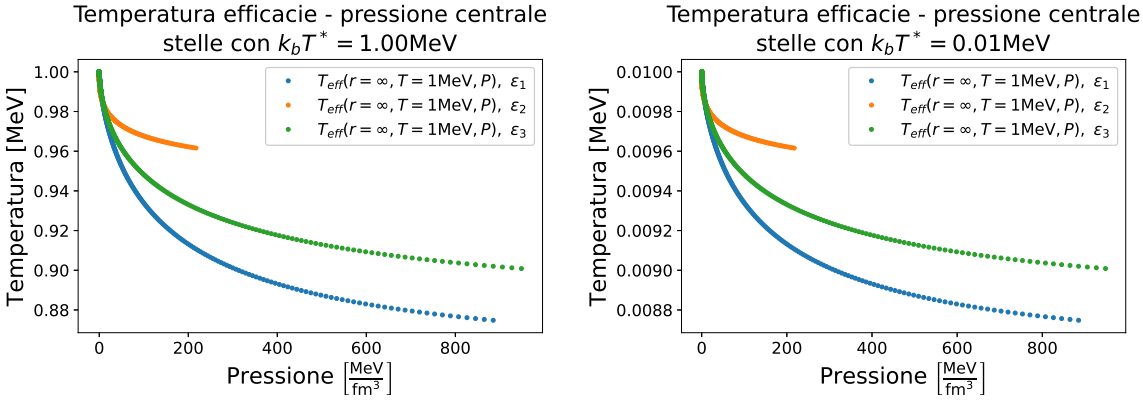

Figura 10: Fissata la temperatura della stella ($\hat{T} = 1$ a sinistra e $\hat{T} = 0.01$ a destra) è possibile esprimere la temperatura misurata all'infinito in funzione della pressione centrale che, data una politropica, ne determina la massa.

A RK4

Codice con cui è stato implementato il metodo `RK4`. `fun_P()` e `fun_m()` sono le funzioni presenti a destra dell'uguale nella prima e nella seconda riga del sistema 5.

```
void rungeKutta4(double h, double r, double *P, double *m, int tipo_politropica){

    double k1, k2, k3, k4, l1, l2, l3, l4;

    k1 = h * fun_m(r, *P, tipo_politropica);
    l1 = h * fun_P(r, *P, *m, tipo_politropica);

    k2 = h * fun_m(r + h / 2, *P + l1 / 2, tipo_politropica);
    l2 = h * fun_P(r + h / 2, *P + l1 / 2, *m + k1 / 2, tipo_politropica);

    k3 = h * fun_m(r + h / 2, *P + l2 / 2, tipo_politropica);
    l3 = h * fun_P(r + h / 2, *P + l2 / 2, *m + k2 / 2, tipo_politropica);

    k4 = h * fun_m(r + h, *P + l3, tipo_politropica);
    l4 = h * fun_P(r + h, *P + l3, *m + k3, tipo_politropica);

    *m += (k1 + 2 * k2 + 2 * k3 + k4) / 6;
    *P += (l1 + 2 * l2 + 2 * l3 + l4) / 6;
}
```

B Risoluzione numerica di $\hat{P}(\rho)$

Quando si calcola il valore della derivata di P o m ad un dato r serve anche il valore dell'energia interna, infatti

```
// f_m = r^2 E
double fun_m(double r, double P, int tipo_politropica){
    return r * r * fun_E(P, tipo_politropica);
}


// f_P = - (P + E)(m + r^3 P)/(r^2 - 2mr)
double fun_P(double r, double P, double m, int tipo_politropica){
    if (m == 0)
        return 0;
    return (P + fun_E(P, tipo_politropica)) * (m + pow(r, 3) * P)
           / ((2 * m - r) * r);
}
```

Dove la funzione `fun_E()` è definita come

```
double fun_E(double P, int tipo_politropica){

    // Politropica quasi realistica (rho*mc^2 + a*rho^alpha + b*rho^beta)
    if (tipo_politropica == 1){
        double rho = findRho(P);
        return rho + A * pow(rho, ALPHA + 1.) + B * pow(rho, BETA + 1.);
    }

    double lambda, K;

    // Materia fermionica non relativistica
    if (tipo_politropica == 2){
        lambda = 5. / 3.;
        K = 0.05;
    }
    else if (tipo_politropica == 3){
        lambda = 2.54;
```

```
18          K = 0.01;
19      }
20      else {
21          printf("Tipo politropica non riconosciuto\n");
22          return 0;
23      }
24
25      double a1 = P / (lambda - 1.);
26      return a1 + pow(a1 / K, 1. / lambda);
27  }
```

Per le politropiche semplici (eq. 2b) abbiamo potuto trovare un'espressione analitica per $\rho(P)$ (eq. 4b). L'energia può quindi essere calcolata in modo diretto come viene fatto nelle righe 22-23 del codice sopra riportato.

Per la politropica 2a, la relazione tra ρ e P che si trova è data in 4a, che riportiamo

$$P = (\alpha - 1)a \left(\frac{n}{n_0}\right)^{\alpha} + (\beta - 1)b \left(\frac{n}{n_0}\right)^{\beta} . \tag{32}$$

Data una certa pressione P bisopgna quindi risolvere numericamente l'equazione per trovare il valore n che la soddisfa. Per fare ciò utiliziamo la funzione `findRho()` così definita

```
1   double findRho(double P){
2
3       // Newton-Raphson method to find rho
4       double rho = pow(P / 0.15, 1. / 3.);     // buona approssimazione iniziale
5
6       while (fabs(P - P_of_rho(rho)) > 1e-8){
7           rho -= (P_of_rho(rho) - P) / DP_of_rho(rho);
8       }
9       return rho;
10  }
```

dove `P_odf_rho()` e `DP_of_rho()` sono rispettivamente la funzione 32 e la sua derivata. In questo modo possiamo sempre trasformare $\epsilon(\rho)$ in $\epsilon(P)$.

C Risoluzione dell'integrale del potenziale gravitazionale

Per ognuna delle 3 stelle trovate (la più massiva per ogni diversa equazione di stato) salviamo ogni valore di r, m e P in un file che successivamente importiamo (riga 3) e utiliziamo per calcolare l'integrale

```
1   for (int tipo_politropica = 1; tipo_politropica < 4; tipo_politropica++){
2       int lenfile = len_files[tipo_politropica - 1];
3
4       double r[lenfile], P[lenfile], m[lenfile], Phi[lenfile];
5
6       read_maxM_data(tipo_politropica, lenfile, r, P, m);
7
8       double R = r[lenfile - 1];
9       double M = m[lenfile - 1];
10      double Phi_ext = fun_Phi_ext(R, M);
11      double integral = 0;
12      double h = 1e-5;
13
14      // Partiamo a calcolare Phi dalla fine (r = R) perche' e' quando
15      // l'integrale e' piu' piccolo
16      for (int i = lenfile - 1; i > 0; i--){
17          integral += h / 2 * (fun_to_integrate(r[i], m[i], P[i])
18                      + fun_to_integrate(r[i - 1], m[i - 1], P[i - 1]));
19          Phi[i] = Phi_ext + integral;
20      }
```

```
21
22      char Phi_int_filename[50];
23      sprintf(Phi_int_filename, "../data/Phi_int_%d.csv", tipo_politropica);
24      FILE *f_Phi_int = fopen(Phi_int_filename, "w");
25      fprintf(f_Phi_int, "r,Phi\n");
26      for (int i = 0; i < lenfile; i++)
27          fprintf(f_Phi_int, "%.10e,%.10e\n", r[i] * R0, Phi[i]);
28      fclose(f_Phi_int);
29
30      // Calcoliamo anche Phi_ext(r) per r > R
31      double r_ext = R;
32
33      char Phi_ext_filename[50];
34      sprintf(Phi_ext_filename, "../data/Phi_ext_%d.csv", tipo_politropica);
35      FILE *f_Phi_ext = fopen(Phi_ext_filename, "w");
36      fprintf(f_Phi_ext, "r,Phi\n");
37
38      while (r_ext < 150 / R0){
39          fprintf(f_Phi_ext, "%.10e,%.10e\n",
40                  r_ext * R0, fun_Phi_ext(r_ext, M));
41          r_ext += h;
42      }
43
44      fclose(f_Phi_ext);
45  }
```

L'integrale (calcolato esplicitamente nella righe 17-18) può essere valutato solo nei punti r che sono stati utilizzati durante la risoluzione delle equazioni di stabilità della stella.

Per ottimizzare il codice, invece che calcolare l'intero integrale per ogni punto r del grafico di $\Phi(r)$, partiamo da $r = R$ (ovvero quando l'integrale è nullo) e aggiungiamo il valore di 1 trapezio per volta alla variabile `integral`. In contemporanea, durante una iterazione del ciclo `for` possiamo utilizzare il valore (parziale) di `integral` per calcolare Φ_{int} in r.

Infine dalla riga 30 calcoliamo Φ_{ext} utilizzando la funzione analitica.

D Integrali con trapezi e simpson

Le funzioni che fanno l'integrale con i due metodi sono

```
1   // Metodo Trapezi
2   double integrale_trapezio(double a, double b, int N, double T,
3                                                   double (*fun)(double, double)){
4       // assume a < b
5       double h = (b - a)/ (double)N;
6       double integral = ((*fun)(a, T) + (*fun)(b, T)) * h / 2;
7
8       for (int i = 1; i < N; i++) {
9       integral += h * (*fun)(a + i * h, T);
10      }
11
12      return integral;
13  }
14
15  // Metodo Simpson
16  double integrale_simpson(double a, double b, int N, double T,
17                                                  double (*fun)(double, double)){
18      // assume a < b
19      double h = (b - a)/N;
20      double integral = ((*fun)(a, T) + (*fun)(b, T)) * h / 3;
21
22      // assume N pari
23      if (N % 2 != 0){
24          printf("N non e' pari");
25      }
26      for (int i = 1; i <= N/2 - 1; i++) {
27          integral += (2 * h /3) * (*fun)(a + 2 * i * h, T);
```

```
        integral += (4 * h / 3) * (*fun)(a + (2 * i - 1) * h, T);
    }

    integral += (4 * h / 3 ) * (*fun)(a + (N - 1) * h, T);

    return integral;
}
```

La funzione da integrare (la radianza) è stata definita come

```
double funB(double nu, double T){
    return pow(nu, 3) / (exp(nu / T) - 1.);
}
```

Per controllare la convergenza del valore della potenza usiamo inizialmente `nu_max` (ovvero $\hat{\nu}_{\max}$) uguale a 20. Verifichiamo per quale `N` si ottiene un errore minore di quello voluto prima per i trapezi e poi per Simpson. I risultati vengono stampati sul terminale.

```
int N = 10;
int N_trap, N_simp;
double Pot, nu_max = 20.;
double Pot_cvg = integrale_trapezio(1e-12, nu_max, 1e8, 1., &funB);
double errore_max = 1e-7;
printf("Errore massimo scelto = %.0e\n", errore_max);
int kk = 0;

// Dati per grafico cvg per N trapezi
FILE *f0 = fopen("../data/potenza/test_cvg_N_trap.csv", "w");
fprintf(f0, "I,N,nu_max\n");
while(N <= 1e6){
    Pot = integrale_trapezio(1e-12, nu_max, N, 1., &funB);
    fprintf(f0, "%.13e,%d,%.13e\n", Pot, N, nu_max * nu0);

    if ((fabs(Pot - Pot_cvg) / Pot_cvg) < errore_max && kk == 0){
        printf("Per trapezi N = %d e' sufficiente\n", N);
        N_trap = N;
        kk++;
    }

    N *= 1.1;
    if (N % 2 != 0) N += 1;
}
fprintf(f0, "%.13e,%d,%.13e\n", Pot_cvg, (int)1e8, nu_max * nu0);
fclose(f0);


N = 10;
kk = 0;
// Dati per grafico cvg per N simpson
FILE *f1 = fopen("../data/potenza/test_cvg_N_simp.csv", "w");
fprintf(f1, "Pot,N,nu_max\n");
while(N <= 1e6){
    Pot = integrale_simpson(1e-12, nu_max, N, 1., &funB);
    fprintf(f1, "%.13e,%d,%.13e\n", Pot, N, nu_max * nu0);

    if ((fabs(Pot - Pot_cvg) / Pot_cvg) < errore_max && kk == 0){
        printf("Per Simpson N = %d e' sufficiente\n", N);
        N_simp = N;
        kk++;
    }

    N *= 1.1;
    if (N % 2 != 0) N += 1;
}
fprintf(f1, "%.13e,%d,%.13e\n", Pot_cvg, (int)1e8, nu_max * nu0);
fclose(f1);
```

```
49
50 // Decidiamo che va bene N_trap e N_simp risultati del codice sopra
51 // Ora verifichiamo la convergenza di nu_max nel caso peggiore, ovvero
52 Pot_cvg = integrale_trapezio(1e-12, 200, 1e8, 1., &funB);
53 nu_max = 20.;
54 int Nrel_trap = (double)N_trap / nu_max; // Teniamo la stessa densita'
55 int Nrel_simp = (double)N_simp / nu_max; // di trapezi: N / nu_max
56
57 // partiamo da
58 nu_max = 10.;
59 kk = 0;
60 // Test cvg per A (ovvero nu_max)
61 FILE *f2 = fopen("../data/potenza/test_cvg_A_trap.csv", "w");
62 fprintf(f2, "Pot,N,nu_max[ad]\n");
63 while(nu_max <= 1e2){
64     N = Nrel_trap * nu_max;
65     if (N % 2 != 0) N += 1;
66
67     Pot = integrale_trapezio(1e-12, nu_max, N, 1., &funB);
68     fprintf(f2, "%.13e,%d,%.13e\n", Pot, N, nu_max);
69
70     if ((fabs(Pot - Pot_cvg) / Pot_cvg) < errore_max && kk == 0){
71         printf("Per trapezi N = %d, nu_max = %.7f sono sufficienti\n",
72                                                                 N, nu_max);
73         kk++;
74     }
75
76     nu_max *= 1.05;
77 }
78 fprintf(f2, "%.13e,%d,%.13e\n", Pot_cvg, (int)1e8, 200.);
79 fclose(f2);
80
81 nu_max = 10.;
82 kk = 0;
83 // Test cvg per A (ovvero nu_max)
84 FILE *f3 = fopen("../data/potenza/test_cvg_A_simp.csv", "w");
85 fprintf(f2, "Pot,N,nu_max[ad]\n");
86 while(nu_max <= 1e2){
87     N = Nrel_simp * nu_max;
88     if (N % 2 != 0) N += 1;
89
90     Pot = integrale_simpson(1e-12, nu_max, N, 1., &funB);
91     fprintf(f3, "%.13e,%d,%.13e\n", Pot, N, nu_max);
92
93     if ((fabs(Pot - Pot_cvg) / Pot_cvg) < errore_max && kk == 0){
94         printf("Per Simpson N = %d, nu_max = %.7f sono sufficienti\n",
95                                                                 N, nu_max);
96         kk++;
97     }
98
99     nu_max *= 1.05;
100 }
101 fprintf(f3, "%.13e,%d,%.13e\n", Pot_cvg, (int)1e8, 200.);
102 fclose(f3);
```

In `stdout` otteniamo

```
Errore massimo scelto = 1e-07
Per trapezi N = 196 e' sufficiente
Per Simpson N = 262 e' sufficiente
Per trapezi N = 264, nu_max = 29.2526072 sono sufficienti
Per Simpson N = 312, nu_max = 24.0661923 sono sufficienti
```

Dalla riga 50 viene utilizzato un codice molto simile a quello sopra controllare la convergenza per diversi valori di `nu_max`. L'unica differenza sta nel fatto che ci assicuriamo che la densità di trapezi `N / nu_max` rimanga costante al variare di `nu_max`.

E Calcolo Temperatura efficace

Per calcolare la temperatura efficace si riutilizzano molte delle funzioni scritte per l'appendice D. Per fare gli integrali possiamo affidarci ai parametri `N_trap` e `nu_max` trovati in D poiché sono stati studiati apposta per valere nel intervallo di temperature che va da 0.1 a 1.

Vista l'equivalenza dei due metodi di integrazione in quanto a precisione, ma non in quanto a costo computazionale, per questo calcolo utilizziamo solamente il metodo dei trapezi.

Il codice è il seguente:

```
#define PI 3.1415926535      // \pi

int N_trap = 22936;
double nu_max = 24.0661923;
double T_min = 0.01;
double T_max = 1;

double R[3] = {11.04289 / R0, 10.86752 / R0, 8.531525 / R0}; // Raggi stelle
double M[3] = {2.456841 / M0, 0.990100 / M0, 1.635845 / M0}; // Masse stelle


// ciclo sulle stelle
for (int i = 0; i < 3; i++){
    char filename[50]; sprintf(filename, "../data/potenza/Teff_%d.csv", i + 1);
    FILE *f = fopen(filename, "w");
    fprintf(f, "T,Teff\n");

    double T = T_min;
    double Integrale, Teff;

    while (T <= T_max){
        Integrale = integrale_trapezio(1e-12, nu_max, N_trap, T, &funB);
        Teff = pow(15, 1. / 4.) / PI * pow(1. - 2. * M[i] / R[i], 1. / 8.);
        Teff *= pow(Integrale, 1. / 4.);
        fprintf(f, "%.7e,%.7e\n", T, Teff);
        T += 0.001;
    }
    fclose(f);
}
```