Relazione di Tirocinio

Leonardo Cruciani

supervised by Prof. Luca Tornatore

July 14, 2019

1 Introduzione teorica

Nel contesto dello studio delle *mass functions* (MF), una buona approssimazione del comportamento degli N-body è data, sotto opportune ipotesi, dalle funzioni di collasso ellissoidale (???)

2 Interpolazione trilineare

Per il caso di interpolazione lineare si vede facilmente che un punto $(x, f(x))$ che giace sulla retta passante per $(x_0, f(x_0))$ e $(x_1, f(x_1))$ soddisfa la relazione:

$$\frac{f(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Da cui si ottiene:

$$f(x) = \frac{1}{x_1 - x_0}\left((x - x_0)f(x_1) + (x_1 - x)f(x_0)\right)$$

La generalizzazione al caso bi- e tri- lineare consiste nella reiterazione del procedimento per ciascuna variabile. Introducendo l'interpolante lineare lungo $\hat{x}$:

$$g(x, y_i) = \frac{1}{x_1 - x_0}\left((x - x_0)f(x_1, y_i) + (x_1 - x)f(x_0, y_i)\right)$$

Si ottiene (per semplicità di notazione $f_{ij} = f(x_i, y_j)$ dove $i, j = 0, 1$):

$$\begin{aligned} f(x,y) &= \frac{1}{y_1 - y_0}\left((y - y_0)g(x, y_1) + (y_1 - y)g(x, y_0)\right) = \\ &= \frac{1}{(x_1 - x_0)(y_1 - y_0)}\big[(y - y_0)(x - x_0)f_{11} + (y - y_0)(x_1 - x)f_{01} + \\ &+ (y_1 - y)(x - x_0)f_{10} + (y_1 - y)(x_1 - x)f_{00}\big] = \\ &= \frac{1}{\Delta\sigma}\begin{bmatrix} x - x_0 & x - x_1 \end{bmatrix} \cdot \begin{bmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{bmatrix} \cdot \begin{bmatrix} y - y_0 \\ y_1 - y \end{bmatrix} = \frac{1}{\Delta\sigma}\bar{x}^\intercal \cdot \bar{\bar{F}} \cdot \bar{y} \end{aligned}$$

Con una conveniente notazione vettoriale.
Si procede come prima con:

$$h(z_i) = \frac{1}{(x_1 - x_0)(y_1 - y_0)}\bar{x}^\intercal \cdot \bar{\bar{\bar{F}}}(z_i) \cdot \bar{y}$$

In tal modo:

$$\begin{aligned} f(x,y,z) &= \frac{1}{z_1 - z_0}\left((z - z_0)h(x, y, z_1) + (z_1 - z)h(x, y, z_0)\right) = \\ &= \frac{1}{(x_1 - x_0)(y_1 - y_0)(z_1 - z_0)}\left[(z - z_0)\,\bar{x}^\intercal \cdot \bar{\bar{F}}(z_1) \cdot \bar{y} + (z - z_0)\,\bar{x}^\intercal \cdot \bar{\bar{F}}(z_0) \cdot \bar{y}\right] = \\ &= \frac{1}{\Delta\tau}\bar{x}^\intercal \cdot \left[(z - z_0)\bar{\bar{F}}(z_1) + (z - z_0)\bar{\bar{F}}(z_1)\right] \cdot \bar{y} = \frac{1}{\Delta\tau}\bar{x}^\intercal \cdot \left[\bar{z} \otimes \bar{\bar{\bar{F}}}\right] \cdot \bar{y} \end{aligned}$$

Notazione scalare: svolgendo i prodotti si ottiene la seguente interpolazione di f, che verrà manipolata al fine di facilitarne l'implementazione in un codice :

$$f(x,y,z) = \sum_{i,j,k}^{0,1} \tilde{x}_i\, \tilde{y}_j\, \tilde{z}_k\, f(x_i, y_j, z_k) \qquad \begin{cases} \tilde{x}_0 = x_1 - x \\ \tilde{x}_1 = x - x_0 \end{cases}$$

Il vettore a due componenti $\tilde{x}$ può essere rappresentato più concisamente con:

$$\tilde{x}[i] = (-)^{i-1} \left(x - x_{|i-1|}\right)$$

Si può quindi applicare la seguente notazione anche alle componenti y e z ed introdurre tre liste $x_0[l]$, $y_0[m]$ e $z_0[n]$ contenenti gli estremi degli intervalli in cui sono stati divisi gli assi.
Per identificare ciascuno dei parallelepipedi che compongono il dominio 3D di f saranno neccessari tutti e tre gli indici l, m, n.
L'implementazione è quindi immediata, una volta stabilito che parallelepipedo $P(l,m,n)$ appartiene il punto (x,y,z) nel quale si vuole stimare il valore di f, il programma procede a calcolare:

$$R = \sum_{i,j,k}^{0,1} \big(-\big)^{i+j+k+1} \big(x - x_0[l + (i+1) \bmod 2]\big)\big(y - y_0[m + (j+1) \bmod 2]\big)\big(z - z_0[n + (k+1) \bmod 2]\big)\, T$$

Dove i valori assoluti $|i-1|$, la cui sola funzione è quella di restituire 1 per $i = 0$ e 0 per $i = 1$, sono stati sostituiti con la più efficiente operazione *modulo 2* (in python: %2), cioè l'operatore che restituisce il resto della divisione per 2.
I valori tabulabili T sono:

$$T = f\big(x_0[l+i], y_0[m+j], z_0[n+k]\big)$$

3 Test

Il programma è stato testato con le seguenti funzioni:

	$f(x,y,z)$	dominio	no. di divisioni
(1)	$3x + 17y - 123z$	$[-50,50] \times [-50,50] \times [-50,50]$	$1000 \times 1000 \times 1000$
(2)	$7x + 5y + 3z$	$[0,10] \times [-10,10] \times [0,15]$	$2 \times 3 \times 5$
(3)	$T(x) + T(3y) + 5T(z)$	$[-0.1,10] \times [-0.1,10] \times [-0.1,10]$	$10 \times 10 \times 10$
(4)	$x^{1/3} e^{-y^2 - z^2}$	$[-50,50] \times [-50,50] \times [-50,50]$	$1000 \times 1000 \times 1000$
(5)	$\sin(3x+7y) + \sin(13xy) + \sin(z^2)$	$[-10,10] \times [-10,10] \times [-10,10]$	$10000 \times 10000 \times 10000$

Dove la funzione T rappresenta l'onda triangolare di periodo 1:

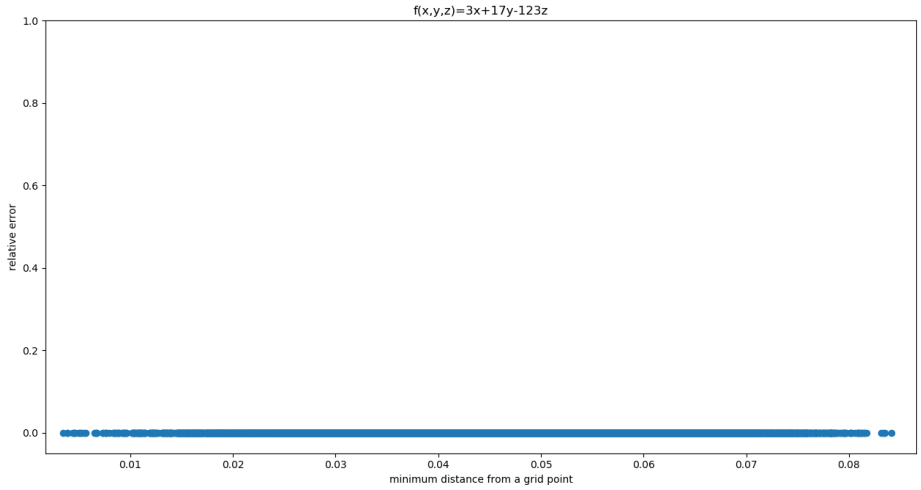
$$T(x) = \frac{|\operatorname{atan}(\tan(\pi x))|}{\pi}$$

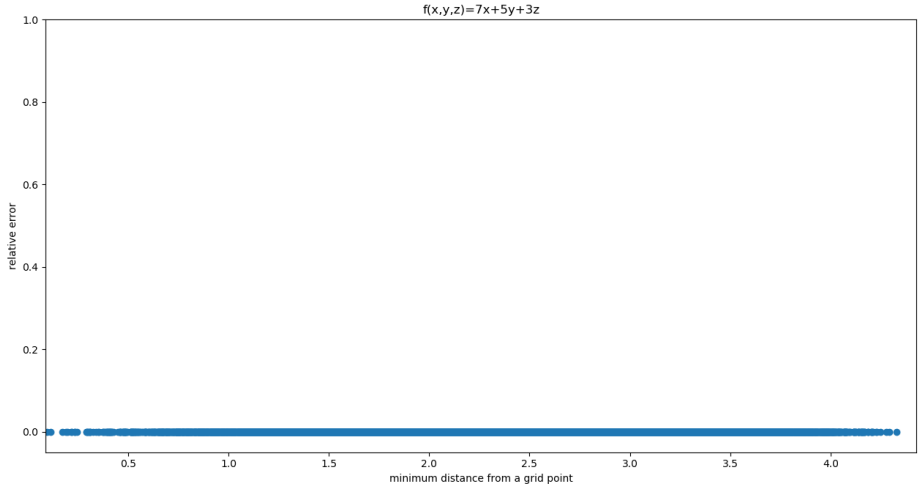
Verificando in particolare che:

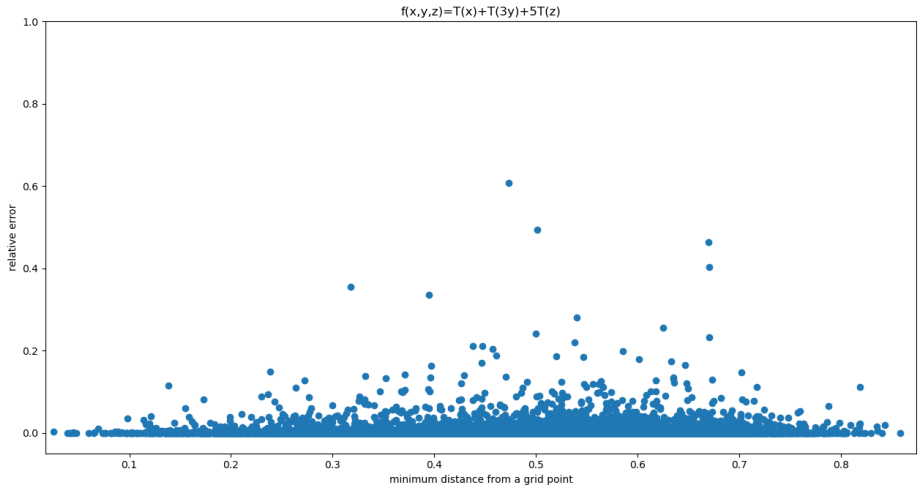
1. Il valore restituito dall'interpolante per i punti griglia coincide con quello reale della funzione (a meno di errori di roundoff)
2. Per le funzioni lineari (1) e (2) il valore coincide con quello vero anche per punti interni, per (3) che ciò avvenga quantomeno non in prossimità di punti spigolosi

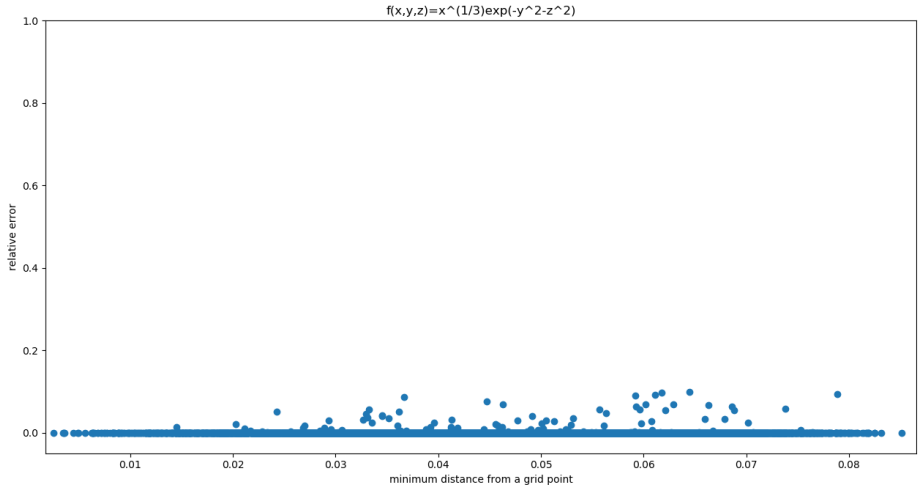
3. Il comportamento del programma è stabile e coerente per punti fuori dal dominio (il programma segnala l'errore e restituisce sempre $R = 0$)

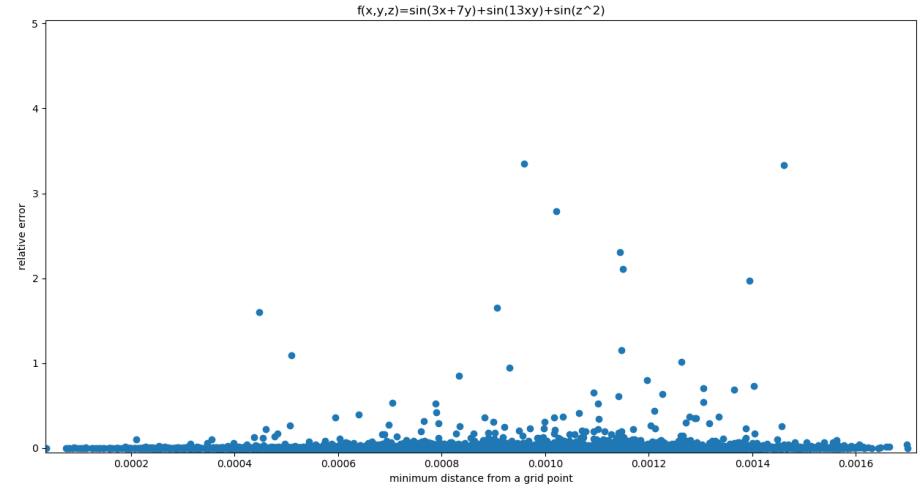
Per valutare l'andamento dell'errore in funzione della distanza, per ciascuna funzione sono stati plottati gli errori relativi in funzione della distanza dal punto griglia più vicino per 10000 punti generati casualmente con distribuzione uniforme sul volume.
Sono riportati di seguito i grafici ottenuti:






f(x,y,z)=T(x)+T(3y)+5T(z)
relative error
minimum distance from a grid point


f(x,y,z)=x^(1/3)exp(-y^2-z^2)
relative error
minimum distance from a grid point


f(x,y,z)=sin(3x+7y)+sin(13xy)+sin(z^2)
relative error
minimum distance from a grid point

Inoltre, per valutare l'andamento dell'errore in funzione del numero di divisioni, per la funzione (4) con dominio ridotto a $[0, 10] \times [0, 10] \times [0, 10]$ è stato plottato l'andamento dell'errore relativo medio $\bar{\varepsilon}$ dei 10 punti riportati in tabella al crescere del numero di divisioni degli assi d (stesso numero di divisioni per tutti e tre gli assi).
Poco sorprendentemente, $\bar{\varepsilon} \to 0$ per $d \to \infty$, con fluttuazioni man mano attenuate probabilmente dovute alla vicinanza dei punti in tabella ai punti griglia per ciascun d.

x	0.50	1.00	1.50	2.00	3.00	4.00	5.00	6.00	7.50	9.00
y	0.10	0.20	0.45	0.75	0.90	1.00	1.25	1.50	1.75	2.00
z	0.10	0.20	0.45	0.75	0.90	1.00	1.25	1.50	1.75	2.00

Di seguito il grafico ottenuto: