Appunti per l'incontro del laboratorio di "Fisica, Matematica e Musica", Trento, Liceo L. da Vinci, 23/2/2006

Sisto Baldo, Dip. Mat. UniTN

La volta scorsa ci siamo posti il problema di approssimare, nel modo migliore possibile, una funzione periodica di periodo 2π (che possiamo pensare come "legge oraria" di una nota musicale emessa da uno strumento) con somme di seni e coseni con periodo sottomultiplo di 2π...

Vediamo di esprimerci in termini un po' più precisi: supponiamo di avere una funzione f, periodica di periodo 2π, e fissiamo un numero naturale N (il numero di *armoniche* che vogliamo usare). Vogliamo scrivere una funzione del tipo

$$f_N(x) = \frac{a_0}{2} + a_1 \cos t + b_1 \sin t + a_2 \cos 2t + b_2 \sin 2t + \ldots + a_N \cos Nt + b_N \sin Nt$$

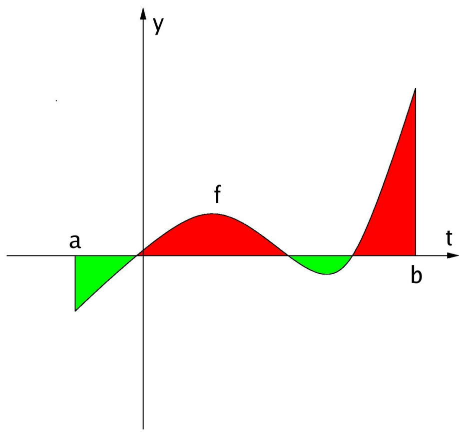
(somma di N armoniche) che approssimi *il meglio possibile* la funzione f. Una funzione come la f_N si chiama *polinomio trigonometrico di grado N*.

La volta scorsa, lavorando su un esempio concreto, abbiamo cercato di stabilire una misura quantitativa di "quanto è buona l'approssimazione": precisamente, siamo stati in grado di definire una distanza tra funzioni che scimmiotta la distanza euclidea nel piano e nello spazio. La nostra definizione era la seguente: la *distanza euclidea* tra le funzioni f_N e f è data da

$$d(f, f_N) = \sqrt{\int_{-\pi}^{\pi} (f(t) - f_N(t))^2},$$

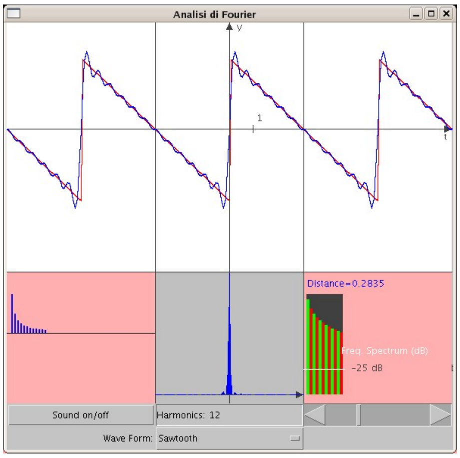
dove $\int_{-\pi}^{\pi}(f(t) - f_N(t))^2$ è l'*integrale* della funzione $(f_N - f)^2$ tra $-\pi$ e π: questo a sua volta è solo un modo un po' pomposo di riferirsi all'area catturata tra il grafico della funzione e l'asse delle t, per t compreso tra $-\pi$ e π.

L'integrale si definisce anche per una funzione il cui grafico non stia tutto *sopra* l'asse delle t: in tal caso, conveniamo che le aree dei "pezzi di sottografico" che stanno sotto l'asse delle t vengono contabilizzati col segno meno: per esempio nella figura che segue



si ha
$$\int_a^b f(t) = \text{Area}(\textit{Regione rossa}) - \text{Area}(\textit{Regione verde}).$$

La volta scorsa abbiamo visto che la nostra definizione di distanza rende conto del fatto che le funzioni $f_n(t) = \sin t + \frac{1}{2}\sin 2t + \frac{1}{3}\sin 3t + \ldots + \frac{1}{n}\sin nt$ si avvicinano sempre di più alla funzione *a dente di sega*: questo avviene precisamente nel senso che, all'aumentare del numero di armoniche n, la *distanza* tra f_n e la funzione a dente di sega f può essere resa piccola quanto si vuole: ecco un'immagine del mio programmino java[1] che visualizza precisamente questo fatto:



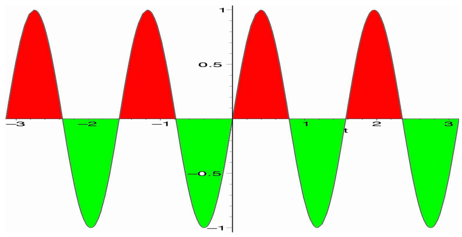
Visto che siamo stati costretti ad introdurre la nozione di integrale, sareste disposti a svolgere insieme a me alcuni semplici "esercizietti" che saranno importantissimi per quello che segue?

[1] *http://www.science.unitn.it/~baldo/divulgazione/Fourier_bis/fourier.html*

Innanzitutto, secondo voi quanto valgono i seguenti integrali?

$$\int_{-\pi}^{\pi} \cos nt, \qquad \int_{-\pi}^{\pi} \sin nt.$$

Osserviamo per esempio la seguente figura, che visualizza il grafico della funzione $\sin 4t$:

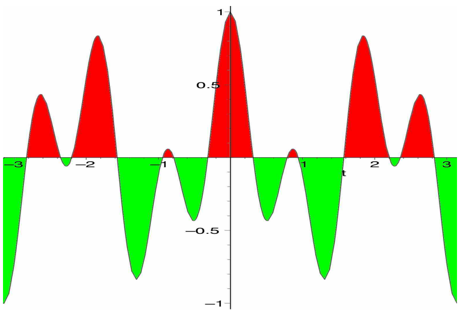


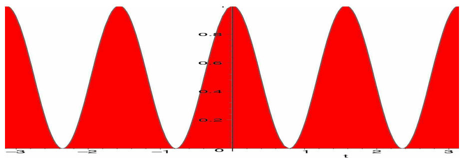
Siccome le aree rosse (positive) sono uguali alle aree verdi (negative), l'integrale cercato vale 0: questo è vero sia per la funzione $\sin nt$ che per la funzione $\cos nt$.

Sempre più difficile: quanto varrà l'integrale

$$\int_{-\pi}^{\pi} \cos nt \sin mt, \quad m, n = 1, 2, 3, \ldots?$$

Questo non è così semplice da indovinare: i due grafici seguenti sono quelli delle funzioni $\cos 2t \cos 5t$ e $\cos 2t \cos 2t = \cos^2 2t$





Osservando le simmetrie, non è difficile indovinare che l'integrale della prima funzione si annulla, mentre quello della seconda è certamente positivo (si tratta di una funzione non negativa!). Putroppo in questo modo non andremo molto lontano...occorre trovare un sistema per risolvere il problemino per ogni valore di m e n!

Per farlo, useremo due semplici proprietà dell'integrale che ci saranno utili anche dopo: la prima è che se moltiplichiamo una funzione f per una costante c, l'integrale risulta moltiplicato per la stessa costante. La seconda proprietà afferma che l'integrale della *somma* di due funzioni è la somma degli integrali delle singole funzioni: in formule

$$\int_{-\pi}^{\pi} cf(t) = c\int_{-\pi}^{\pi} f(t) \qquad \int_{-\pi}^{\pi} (f(t)+g(t)) = \int_{-\pi}^{\pi} f(t) + \int_{-\pi}^{\pi} g(t).$$

Entrambe queste proprietà si possono ottenere da semplici considerazioni geometriche sulle aree...fidatevi!

Ora, grazie alle formule di addizione per il coseno (che sono da sempre l'amore della nostra vita, non è vero?) possiamo scrivere:

$$\cos nt \cos mt = \frac{1}{2}\left[\cos(m-n)t + \cos(m+n)t\right]\ dt,$$

da cui

$$\int_{-\pi}^{\pi} \cos nt \cos mt = \frac{1}{2}\left[\int_{-\pi}^{\pi} \cos(m-n)t + \int_{-\pi}^{\pi} \cos(m+n)t\right]\ dt.$$

Per quanto osservato sopra, i due integrali a destra valgono 0: abbiamo visto che $\int_{-\pi}^{\pi} \cos kt = 0$ per $k = 1, 2, \ldots$. C'è però un'importante eccezione: se $m = n$, il primo integrale del membro di destra *non si annulla affatto*: stiamo facendo l'integrale della costante 1, che vale 2π (area del rettangolo).

In conclusione, possiamo dire che

$$(*a) \int_{-\pi}^{\pi} \cos nt \cos mt = \begin{cases} 0 & se\ m \neq n \\ \pi & se\ m = n \end{cases}$$

In maniera del tutto analoga (potete farlo a casa se proprio non state nella pelle...), possiamo verificare che

$$(*b) \qquad \int_{-\pi}^{\pi} \sin(nt)\ \sin(mt) = \begin{cases} \pi & se\ m = n \\ 0 & se\ m \neq n \end{cases}$$

$$(*c) \qquad \int_{-\pi}^{\pi} \cos(nt)\ \sin(mt) = 0.$$

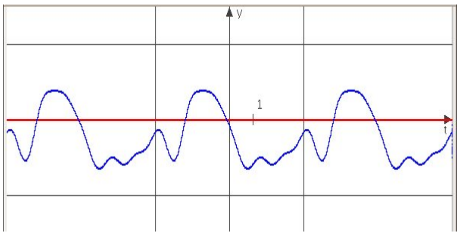
Le tre identità $(*a)$, $(*b)$ e $(*c)$ sono note come *relazioni di ortogonalità del seno e del coseno.*

Dopo aver fatto questo esercizio in verità piuttosto noiosetto, torniamo al nostro problema di trovare il polinomio trigonometrico con n armoniche che meglio approssima una data funzione periodica.

Proviamo per prima cosa a risolvere un problema decisamente più semplice: supponiamo che qualcuno ci passi la "fotografia" del grafico di una funzione $f(t)$, e ci dica che essa è un polinomio trigonometrico ottenuto sommando le funzioni

$$1, \cos t, \sin t, \cos 2t, \sin 2t, \ldots, \cos 6t, \sin 6t$$

con opportuni coefficienti reali che però ha dimenticato:



C'è un modo di *ricostruire i coefficienti* senza procedere per tentativi?

La ricostruzione per tentativi giocando con gli slider dell'applet java sarebbe infatti disperata, perché abbiamo ben 13 coefficienti incogniti da determinare!

Ebbene, la nostra equazione è

$$f(t) = a_0/2 + a_1 \cos t + b_1 \sin t + a_2 \cos 2t + b_2 \sin 2t + \ldots + a_6 \cos 6t + b_6 \sin 6t,$$

con 13 incognite $a_0, a_1, b_0, b_1, \ldots$. Se integriamo ambo i membri tra $-\pi$ e π, e ricordiamo che

$$\int_{-\pi}^{\pi} \sin nt = \int_{-\pi}^{\pi} \cos nt = 0, \ n = 1, 2, 3, \ldots$$

troviamo

$$\int_{-\pi}^{\pi} f(t) = \pi a_0.$$

In modo analogo, e grazie alle relazioni di ortogonalià, troviamo gli altri coefficienti: per esempio, moltiplichiamo l'equazione per $\sin 4t$ e integriamo ambo i membri. Ricordando che $\int_{-\pi}^{\pi} \sin nt \sin 4t = 0$ per $n \neq 4$, mentre $\int_{-\pi}^{\pi} \sin 4t \sin 4t = \pi$ otteniamo $\pi b_4 = \int_{-\pi}^{\pi} f(t) \sin 4t \, dt$, e abbiamo determinato b_4...

Insomma, in generale scopriamo che se

$$f(t) = a_0/2 + a_1 \cos t + b_1 \sin t + a_2 \cos 2t + b_2 \sin 2t + \ldots + a_N \cos Nt + b_N \sin Nt,$$

allora

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos nt \, dt, \quad b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin nt \, dt, \ n = 0, 1, 2, \ldots, N.$$

Siamo cioè in grado di ricostruire il polinomio trigonometrico di partenza, a patto di saper calcolare certi integrali. I coefficienti a_n, b_n calcolati con gli integrali scritti sopra si chiamano *coefficienti di Fourier* di f.

Torniamo infine al problema originale, e supponiamo di prendere una *qualunque* funzione $f(t)$ periodica di periodo 2π. La funzione non dovrà essere eccessivamente schifosa, in modo che sia possibile calcolarne l'integrale.

Ci chiediamo come sia possibile approssimare f *in modo ottimale* usando polinomi trigonometrici di "grado" N (o con N armoniche), cioè con funzioni del tipo

$$(**)\quad g(t) = a_0/2 + a_1 \cos t + b_1 \sin t + a_2 \cos 2t + b_2 \sin 2t + \ldots + a_N \cos Nt + b_N \sin Nt,$$

dove $a_0, a_1, \ldots, a_N, b_1, \ldots, b_N$ sono $2N+1$ coefficienti da scegliere nel modo migliore possibile. E cosa vorrà dire per noi "migliore possibile"? Vogliamo che la *distanza* tra f e g sia la *minima possibile*!

Il discorso sulla ricostruzione dei coefficienti di un polinomio trigonometrico suggerisce che FORSE una buona approssimazione si ottiene scegliendo i coefficienti di Fourier: in altre parole, prendendo quel *particolare polinomio trigonometrico di grado N*

$$\tilde{g}(t) = \tilde{a}_0/2 + \sum_{n=1}^{N} [\tilde{a}_n \cos n + \tilde{b}_n \sin nt],$$

i cui coefficienti sono definiti da

$$(III) \qquad \tilde{a}_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos nt \, dt, \quad \tilde{b}_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin nt \, dt, \qquad n = 0, 1, \ldots, N$$

abbiamo un naturale candidato ad essere una buona approssimazione della funzione f.

In realtà, il polinomio $\tilde{g}$ è proprio *il migliore di tutti*: tra tutti i polinomi trigonometrici di grado N, è quello di *distanza minima* dalla funzione data f.

La cosa bella è che si può raggiungere questo risultato andando *esplicitamente* a minimizzare la distanza, senza dover affatto ricorrere alle "intuizioni" che precedono...Avremo bisogno soltanto della definizione di distanza, delle semplici proprietà dell'integrale che abbiamo visto prima, e delle relazioni di ortogonalità.

Prendiamo dunque un *qualunque* polinomio trigonometrico come in $(**)$, ed andiamo a scrivere la sua distanza (al quadrato) dalla funzione da approssimare f:

$$\begin{aligned} (IV) \quad d(g,f)^2 \quad &= \int_{-\pi}^{\pi} (f(t) - g(t))^2 \, dt = \\ \int_{-\pi}^{\pi} (f(t) - a_0/2 \quad &- \quad a_1 \cos t - a_2 \cos 2t - \ldots - a_N \cos Nt \\ &- \quad b_1 \sin t - b_2 \sin 2t - \ldots - b_N \sin Nt)^2 \, dt \end{aligned}$$

Come "aperitivo", vediamo cosa succede per $N = 1$ (cioè se stiamo tentando di approssimare f con un polinomio trigonometrico del tipo $g(t) = a_0/2 + a_1 \cos t + b_1 \sin t$, cioè con una funzione *sinusoidale pura* alla frequenza fondamentale): ovviamente non possiamo sperare in un'approssimazione troppo buona, ma i conti sono estremamente semplificati in quanto abbiamo solo tre coefficienti...ed avremo ottime indicazioni su come fare nel caso generale!

Facciamo il conto sviluppando il quadrato (senza dimenticarci dei doppi prodotti!) e ricordando che l'integrale della somma è la somma degli integrali, e che le costanti si possono portare fuori dal segno di integrale:

$$
\begin{aligned}
&[d(f,g)]^2 = \int_{-\pi}^{\pi} (f(t) - \frac{a_0}{2} - a_1 \cos t - b_1 \sin t)^2 = \\
&\int_{-\pi}^{\pi} [f(t)^2 + \frac{a_0^2}{4} + a_1^2 \cos^2 t + b_1^2 \sin^2 t - a_0 f(t) - 2a_1 f(t) \cos t + \\
&-2b_1 f(t) \sin t + a_0 a_1 \cos t + a_0 b_1 \sin t + 2a_1 b_1 \sin t \cos t] = \\
&\int_{-\pi}^{\pi} f(t)^2 + \frac{a_0^2}{4} \int_{-\pi}^{\pi} 1 + a_1^2 \int_{-\pi}^{\pi} \cos^2 t + b_1^2 \int_{-\pi}^{\pi} \sin^2 t + \\
&-a_0 \int_{-\pi}^{\pi} f(t) - 2a_1 \int_{-\pi}^{\pi} f(t) \cos t - 2b_1 \int_{-\pi}^{\pi} f(t) \sin t + \\
&+a_0 a_1 \int_{-\pi}^{\pi} \cos t + a_0 b_1 \int_{-\pi}^{\pi} \sin t + 2a_1 b_1 \int_{-\pi}^{\pi} \sin t \cos t
\end{aligned}
$$

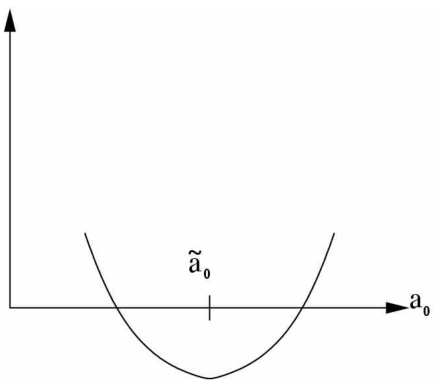
Ora, se ci ricordiamo delle relazioni di ortogonalità e della definizione dei coefficienti di Fourier di f, vediamo che siamo in grado di calcolare tutti e dieci gli integrali che compaiono nell'ultima espressione (tranne il primo che lasciamo scritto allo stesso modo): abbiamo precisamente

$$
\begin{array}{lll}
\int_{-\pi}^{\pi} 1 = 2\pi, & \int_{-\pi}^{\pi} \cos^2 t = \pi, & \int_{-\pi}^{\pi} \sin^2 t = \pi, \\
\int_{-\pi}^{\pi} f(t) = \pi\, \tilde{a}_0, & \int_{-\pi}^{\pi} f(t) \cos t = \pi \tilde{a}_1, & \int_{-\pi}^{\pi} f(t) \sin t = \pi \tilde{b}_1, \\
\int_{-\pi}^{\pi} \cos t = 0, & \int_{-\pi}^{\pi} \sin t = 0, & \int_{-\pi}^{\pi} \sin t \cos t = 0.
\end{array}
$$

Sostituendo nel notro contaccio vediamo quindi che si ha

$$
[d(f,g)]^2 = \int_{-\pi}^{\pi} f(t)^2 + \pi[\frac{1}{2}(a_0^2 - 2a_0\tilde{a}_0) + (a_1^2 - 2a_1\tilde{a}_1) + (b_1^2 - 2b_1\tilde{b}_1)].
$$

Il compito che ci siamo proposti è quello di determinare i valori di a_0, a_1 e b_1 che rendono minima questa distanza, *cioè che rendono minima ciascuna delle tre espressioni tra parentesi tonde nell'ultima espressione!* Ora, non è difficile disegnare il grafico della funzione $(a_0^2 - 2a_0\tilde{a}_0)$ al variare del coefficiente a_0: si tratta di una parabola rivolta verso l'alto



Il vertice della parabola si ha per $a_0 = \tilde{a}_0$, quindi la nostra espressione è minima quando $a_0 = \tilde{a}_0$, e in tal caso vale $-\tilde{a}_0^2$. Analogamente, le altre due espressioni (che guarda caso hanno la stessa forma) sono minime rispettivamente quando $a_1 = \tilde{a}_1$ e $b_1 = \tilde{b}_1$, cioè quando i coefficienti del nostro "polinomio trigonometrico di grado 1" coincidono con i coefficienti di Fourier! E in tal caso la distanza minima vale

$$[d(f, \tilde{g})]^2 = \int_{-\pi}^{\pi} (f(t))^2 - \pi(a_0^2/2 + a_1^2 + b_1^2).$$

Il conto nel caso generale di un polinomio trigonometrico di grado N è sostanzialmente identico: se ci armiamo di estrema pazienza e andiamo a sviluppare il quadrato nell'espressione (IV), usando le relazioni di ortogonalità e le definizioni dei coefficienti di Fourier scopriremo che

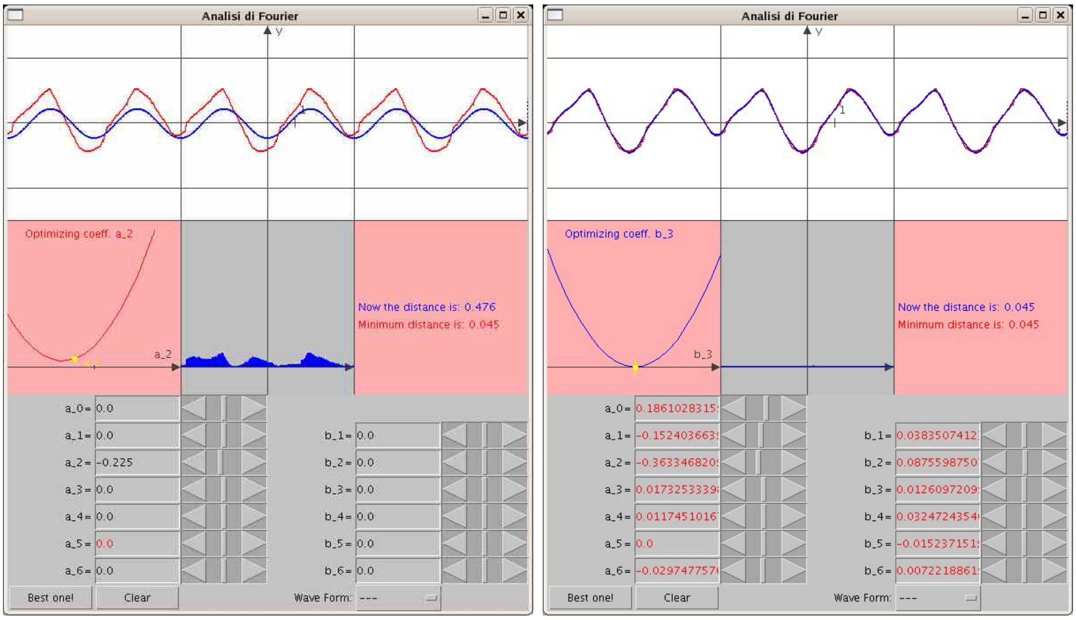
$$(V) \qquad \begin{aligned} d(f, g)^2 &= \int_{-\pi}^{\pi} f(t)^2 \, dt + \\ &\pi[\frac{1}{2}(a_0^2 - 2a_0\tilde{a}_0) + (a_1^2 - 2a_1\tilde{a}_1) + (a_2^2 - 2a_2\tilde{a}_2) + \\ &\ldots + (b_N^2 - 2b_N\tilde{b}_N)]. \end{aligned}$$

Per minimizzare questa distanza ragioniamo esattamente come abbiamo fatto nel caso più semplice, e scopriamo che dobbiamo scegliere nell'ordine $a_0 = \tilde{a}_0$, $a_1 = \tilde{a}_1$,…,$a_N = \tilde{a}_N$, $b_1 = \tilde{b}_1$, …, $b_N = \tilde{b}_N$: il polinomio trigonometrico che ha distanza minima da f è proprio il polinomio di Fourier $\tilde{g}$, e questa distanza minima vale

$$d(f, \tilde{g})^2 = \int_{-\pi}^{\pi} f(t)^2 \, dt - \pi[\frac{\tilde{a}_0^2}{2} + \tilde{a}_1^2 + \ldots + \tilde{a}_N^2 + \tilde{b}_1^2 + \ldots + \tilde{b}_N^2].$$

La formula che abbiamo trovato ci dice una cosa importante: che è possibile "ottimizzare" ciascun coefficiente in modo indipendente dagli altri, e che il grafico della *distanza al quadrato* in funzione di ciascuno dei coefficienti (lasciando fissi tutti gli altri) è una parabola.

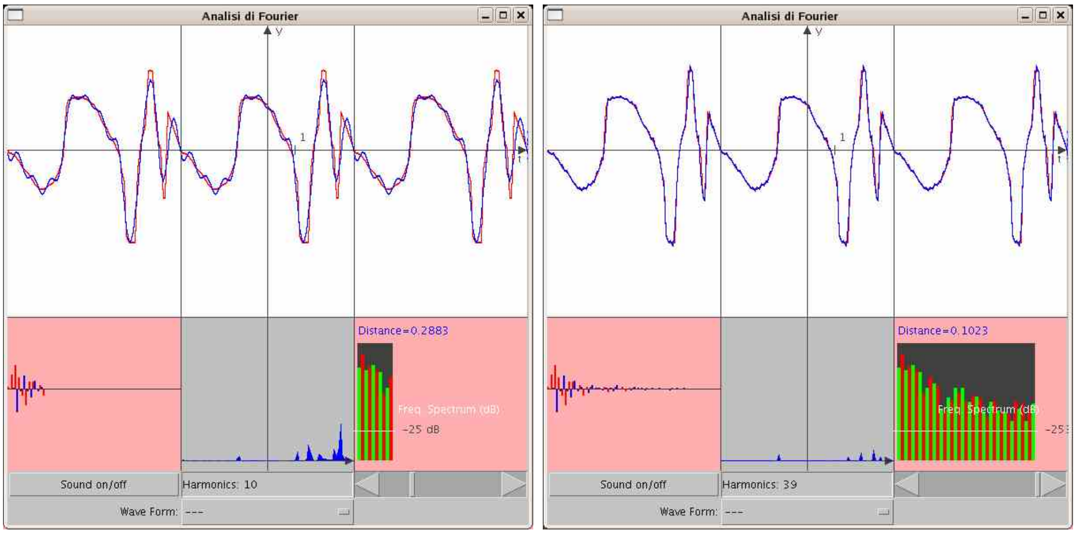
Questo discorso sulla minimizzazione viene reso evidente dall'applet java con gli slider[2]: se spostiamo uno degli slider, la parabola corrispondente a quel coefficiente ci viene appunto mostrata dall'applet (il grafico di sinistra della fascia centrale): basta quindi spostare lo slider finché non siamo sul vertice della parabola, e ripetere il giochetto per tutti i coefficienti.



Dal discorso precedente, è ovvio che all'aumentare di N i polinomi di Fourier daranno approssimazioni sempre migliori (o almeno non peggiori) di f (questo perché un polinomio trigonometrico di grado N può essere visto come un "caso particolare" di polinomio di grado $N + 1$, in cui i coefficienti di $\cos(N + 1)t$ e di $\sin(N + 1)t$ valgono 0).

Un po' di esperimenti con l'applet java mostrano che è ragionevole attendersi che questo processo *converga*, cioè che la distanza tra polinomio trigonometrico approssimante e funzione approssimata possa essere resa *arbitrariamente piccola* scegliendo N *sufficientemente grande*. Nella figura che segue vediamo come l'applet java approssima una funzione da noi tracciata arbitrariamente con il mouse (il grafico rosso), dapprima con un polinomio trigonometrico di grado 10, poi con un polinomio trigonometrico di grado 39 (entrambi tracciati in blu).

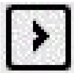
[2] *http://www.science.unitn.it/~baldo/divulgazione/Fourier1/fourier1.html*



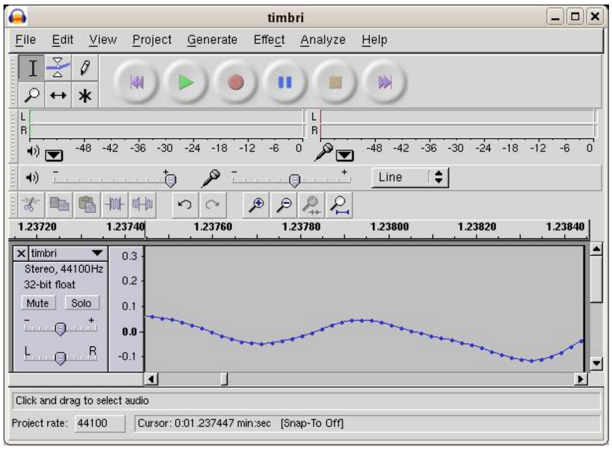
Purtroppo, la *dimostrazione* rigorosa della convergenza del metodo, pur non essendo affatto difficile... richiede un bel po' di teoria elementare degli spazi di Hilbert: se vi iscriverete al Corso di Laurea in Matematica avete buone probabilità di doverla studiare al secondo o al terzo anno!

Tornando all'applet, vale la pena di spiegare il significato degli istogrammi presenti nella parte bassa: quello di sinistra rappresenta i coefficienti di Fourier (le barre rosse rappresentano gli a_n, i coefficienti dei coseni, mentre quelle blu sono i b_n, i coefficienti dei seni). Le barre dell'istogramma di destra rappresentano invece le *ampiezze delle armoniche* (che, come ricorderete, sono date da $A_n = \sqrt{a_n^2 + b_n^2}$) su scala logaritmica: il livello cui arriva una barra è cioè proporzionale al logaritmo dell'ampiezza A_n dell'armonica corrispondente. La bizzarra scelta di una scala logaritmica è dovuta alla fisiologia dell'udito: quel che noi percepiamo come intensità del suono non è proporzionale all'ampiezza dell'onda sonora, ma bensì al suo logaritmo!

Per concludere questa nostra discussione, volevo farvi notare una cosa importante: la teoria che abbiamo sviluppato ci dice che conoscere una forma d'onda è la stessa cosa che conoscere i coefficienti di Fourier della stessa. Anzi, dal punto di vista acustico è sufficiente conoscere le *ampiezze* di un numero sufficiente di armoniche (perché l'orecchio non è sensibile alla fase, né alle frequenze troppo alte)!

Questo fornisce un buon metodo per *comprimere* l'informazione relativa alla forma d'onda per ottenere file piccoli. Mi spiego meglio: in un file musicale *non compresso*, per esempio in formato .wav, la legge oraria del suono viene semplicemente campionata per punti: possiamo rendercene conto aprendo con Audacity il file sonoro, sentito la volta scorsa, che contiene la stessa nota suonata da oboe, violino e tromba: 

Se ingrandiamo a sufficienza il grafico, a un certo punto compaiono dei pallini: essi rappresentano gli istanti temporali per cui il valore della funzione è materialmente “registrato” nel file. Nell’intervallo tra due pallini successivi, viene fatta un’interpolazione!



Per dare un’idea un po’ più precisa dell’*ingombro* di tutta questa informazione, per rappresentare un’oscillazione completa dell’oboe sono necessari circa 300 punti, e per ciascuno dei questi il valore della funzione viene rappresentato come numero intero a 2 bytes ...il tutto moltiplicato per 2 perché l’informazione è duplicata per ciascuno dei due canali stereo. Dunque, l’informazione relativa ad una sola oscillazione completa dell’oboe in un file .wav occupa circa 1.2 KB.

Invece, possiamo prendere la forma d’onda dell’oboe (periodicizzato) e calcolarne i coefficienti di Fourier con un programmino simile all’applet java che abbiamo usato prima (questo programmino “legge” la forma d’onda direttamente dal file .wav, ma calcola i coefficienti di Fourier esattamente come abbiamo imparato, usando un’opportuna approssimazione discreta degli integrali).

I coefficienti calcolati in questo modo sono stati usati per riprodurre i suoni di oboe, violino e tromba (e le vocali AAA e OOO) nel mio applet java: se confrontate con i suoni e le forme d’onda presenti negli appunti dello scorso incontro, vedrete che l’imitazione è più che buona!

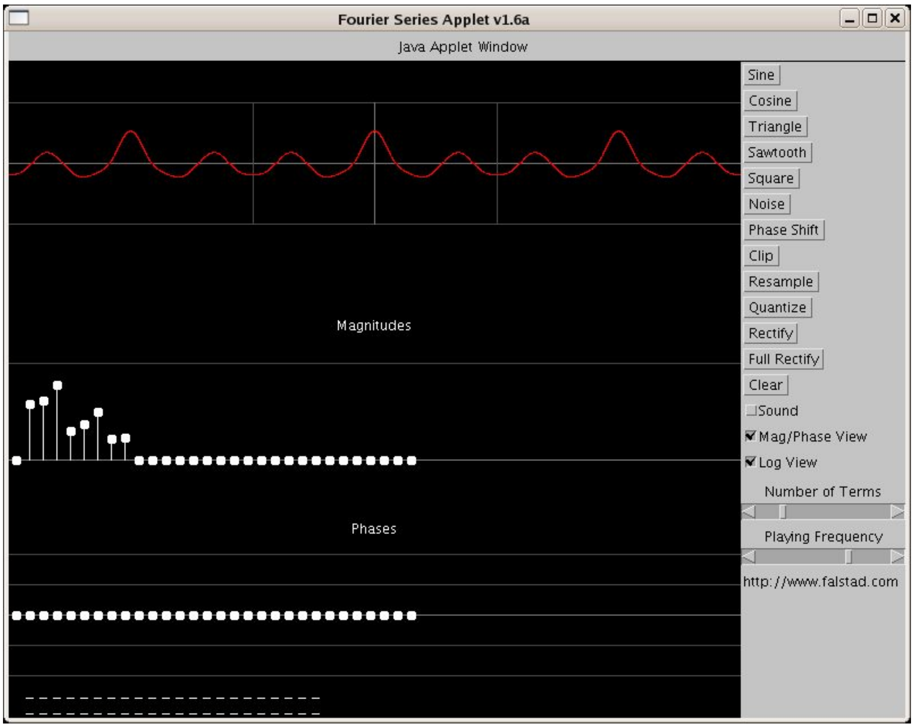
Dai coefficienti di Fourier possiamo anche ricavare l’*ampiezza delle armoniche* per oboe, violino e tromba e per le vocali: le prime 15 armoniche hanno le seguenti ampiezze (espresse in decibel, un’unità di misura legata al logaritmo dell’ampiezza)

OBOE (Ampiezze armoniche in dB):
-11,-10,-6,-18,-16,-13,-20,-20,-27,-28,-33,-32,-24,-32,-34
VIOLINO (Ampiezze armoniche in dB):
-7,-16,-13,-7,-16,-14,-15,-15,-20,-15,-20,-18,-21,-20,-21
TROMBA (Ampiezze armoniche in dB):
-16,-9,-11,-16,-12,-13,-20,-18,-15,-17,-21,-23,-26,-24,-24
AAA (Ampiezze armoniche in dB):
-11,-15,-22,-19,-15,-14,-14,-14,-19,-24,-26,-23,-25,-30,-27
OOO (Ampiezze armoniche in dB):
-10,-11,-13,-19,-15,-16,-27,-28,-34,-28,-26,-24,-31,-31,-32

Quanto spazio ci occorre per registrare in un file l'informazione relativa ad uno degli strumenti? Ciascuno dei nostri numeri può essere immagazzinato in un byte (avanzando anche qualche bit): abbiamo dunque bisogno di 15 bytes in totale! Davvero un bel risparmio di spazio, non trovate?

Ovviamente, per verificare l'efficacia della nostra compressione, occorre vedere se i dati che abbiamo sono sufficienti a ricostruire degnamente il suono originale. Possiamo effettuare la verifica grazie a un altro programmino java[3], scritto da Paul Falstad e disponibile in rete. Selezioniamo nel pannello di destra "Mag/Phase View" e poi "Log View", e premiamo il tasto "Clear". A questo punto siamo liberi di impostare i coefficienti di Fourier in dB di uno dei nostri suoni, per esempio l'oboe: saranno sufficienti i primi 8. Impostiamo poi una frequenza di circa 500Hz e selezioniamo "Sound": riecco, con buona approssimazione, il nostro oboe!

[3] *http://www.falstad.com/fourier/*



Naturalmente la forma d'onda è diversa da quella dell'oboe: abbiamo perso l'informazione sulle fasi!

L'esperimento funziona piuttosto bene anche con le vocali: in questo caso, conviene impostare una frequenza di circa 100Hz.