

# SIM-2NV - Sistemi di intonazione musicale

## Manuale Utente v1.75

Autore: LUCA BIMBI  
Data: 2 Settembre 2025  
Versione: 1.75  
Licenza: MIT (vedi LICENSE)

### Indice

- 1. Introduzione
- 2. Requisiti e Installazione
- 3. Uso Rapido
- 4. Parametri della Riga di Comando
- 5. Sistemi di Accordatura
- 6. File e Tabelle di Output
- 7. Gestione del Range MIDI
- 8. Esempi di Utilizzo
- 9. Note e Limitazioni
- 10. Crediti

### Introduzione

`SIM-2NV.py` è uno strumento a riga di comando per generare sistemi di intonazione e file di supporto per la sintesi e l'analisi musicale. È pensato in particolare per:

- generare tabelle `cpstun` (GEN -2) per Csound;
- esportare file `AnaMark .tun`;
- produrre tabelle testuali ed Excel del sistema generato;
- confrontare le frequenze del sistema con 12-TET, serie armonica e serie subarmonica.

È un compagno del progetto SIM e mantiene la stessa filosofia del manuale generale (`manual.md`), focalizzandosi però sull'eseguibile `SIM-2NV.py` e sulle sue specificità.

### Requisiti e Installazione

- Python 3.6+ (le annotazioni usano `typing.Union`; l'operatore `|` di 3.10+ non è richiesto).
- Moduli standard: `argparse`, `sys`, `math`, `re`, `os`, `shutil`, `fractions`, `typing`.
- Opzionale per export Excel: `openpyxl`.
  - Installazione: `pip install openpyxl`

Installazione/uso locale: - Tenere `SIM-2NV.py` in una cartella accessibile. - Su sistemi Unix-like, opzionale: `chmod +x SIM-2NV.py`.

### Uso Rapido

Formato generale:

- Python: `python SIM-2NV.py [opzioni] OUTPUT_BASE`
- Unix-like (se eseguibile): `./SIM-2NV.py [opzioni] OUTPUT_BASE`

Dove `OUTPUT_BASE` è il nome base dei file generati (senza estensione).

Esempio minimo (12-TET su ottava, default):

```
python SIM-2NV.py out
```

Questo genera: - un file `out.csd` contenente una tabella `cpstun` GEN -2; - tabelle del sistema `out_system.txt` (+ `.xlsx` se `openpyxl` è installato); - tabelle di confronto `out_compare.txt` (+ `.xlsx` se `openpyxl` è installato).

Opzioni utili rapide: - `--export-tun` per generare anche `out.tun`. - `--basenote A4 --diapason 442` per cambiare nota di riferimento e diapason. - `--span 3` per estendere la serie su 3 intervalli di ripetizione.

## Parametri della Riga di Comando

Nota: i default sono tra parentesi.

### Tabelle riepilogative dei parametri CLI

#### Base

Parametro	Argomenti	Tipo input	Obbligatorio	Default	Descrizione
<code>--v, --version</code>	-	flag	No	-	Stampa la versione del programma
<code>--diapason</code>	<Hz>	float (Hz)	No	440.0	Diapason A4 in Hertz
<code>--basekey</code>	<MIDI>	int (0..127)	No	60	Nota MIDI di partenza per la tabella
<code>--basenote</code>	<nome Hz>	nota (es. A4, F#2, Bb3) oppure float (Hz)	No	“C4”	Nota/frequenza di riferimento per i calcoli in Hz

#### Sistemi di intonazione (sceglierne uno; opzionali)

Parametro	Argomenti	Tipo input	Obbligatorio	Default	Descrizione
<code>--et</code>	INDEX INTERVAL	INDEX: int (>0); INTERVAL: int o frazione (cents o frazione convertita in cents)	No	12 1200	Temperamento equabile sull’intervallo specificato
<code>--geometric</code>	GEN STEPS INTERVAL	GEN: int/frazione/float (>0); STEPS: int (>0); INTERVAL: rapporto (float/frazione) oppure cents (int o con suffisso c, cent, cents)	No	-	Progressione geometrica con riduzione nell’intervallo indicato
<code>--natural</code>	A_MAX B_MAX	int, int (>=0)	No	-	Sistema naturale 4:5:6 con riduzione all’ottava salvo <code>--no-reduce</code>
<code>--danielou</code>	a,b,c	tre interi (possono essere negativi; usare virgolette se necessario)	No	-	Aggiunge rapporti del sistema Danielou; opzione ripetibile
<code>--danielou-all</code>	-	flag	No	-	Genera la griglia completa Danielou

#### Opzioni aggiuntive

Parametro	Argomenti	Tipo input	Obbligatorio	Default	Descrizione
<code>--no-reduce</code>	-	flag	No	-	Non riduce all’ottava/intervallo
<code>--span, --ambitus</code>	N	int (>=1)	No	1	Ripete la serie su N intervalli di ampiezza INTERVAL
<code>--interval-zero</code>	-	flag	No	-	Forza <code>interval=0</code> nella tabella cpstun (GEN -2)

Parametro	Argomenti	Tipo input	Obbligatorio	Default	Descrizione
<code>--export-tun</code>	-	flag	No	-	Esporta file AnaMark <code>.tun</code>
<code>--tun-integer</code>	-	flag	No	-	<code>.tun</code> : arrotonda i cents al valore intero più vicino

Confronti

Parametro	Argomenti	Tipo input	Obbligatorio	Default	Descrizione
<code>--compare-fund</code>	[val]	nota/Hz oppure parola chiave <code>basenote</code>	No	<code>basenote</code>	Fondamentale per confronto (opzione utilizzabile anche senza argomento)
<code>--compare-tet-align</code>	{same, nearest}	scelta	No	<code>same</code>	Allineamento della griglia 12TET per il confronto
<code>--subharm-fund</code>	<val>	nota/Hz	No	<code>A5</code>	Fondamentale per la serie subarmonica (se non valido, usa <code>diapason</code> )
<code>--midi-truncate</code>	-	flag	No	-	Troncamento al range MIDI 0..127 anziché adattare <code>basekey</code>

Argomento posizionale

Parametro	Argomenti	Tipo input	Obbligatorio	Default	Descrizione
<code>OUTPUT_BASE</code>	nome base	string	No	<code>out</code>	Base dei nomi file generati (senza estensione)

Base: - `--v`, `--version` — stampa la versione. - `--diapason` <Hz> — diapason A4 in Hz (440.0). - `--basekey` <MIDI> — nota MIDI base per la tabella (60 = C4). - `--basenote` <nome|Hz> — nota/frequenza di riferimento per calcolo Hz (“C4”). - Accetta nomi come `A4`, `F#2`, `Bb3` oppure un numero in Hz (es. 261.625).

Sistemi di intonazione (sceglierne uno per volta; se più opzioni sono presenti, la logica interna seleziona la prima valida nell’ordine qui indicato): - `--natural` `A_MAX` `B_MAX` — sistema naturale 4:5:6, genera i rapporti  $3/2^a$  e  $5/4^b$  combinati, ridotti all’ottava salvo `--no-reduce`. - `--danielou` `a,b,c` — aggiunge rapporti del sistema Danielou derivati da esponenti per 6/5, 3/2,  $2^c$ . Opzione ripetibile. - `--danielou-all` — genera l’intera griglia (53 gradi unici con 2.0 come limite superiore se riduzione attiva). - `--geometric` `GEN` `STEPS` `INTERVAL` — progressione geometrica: - `GEN`: generatore (intero o frazione es. 3/2; ammesso anche float). - `STEPS`: numero di passi (>0). - `INTERVAL`: intervallo di ripetizione; accetta `int/frazione/float` oppure cents con suffissi `c`, `cent`, `cents` (es. 700c). Deve essere >1 (o >0 in cents). - `--et` `INDEX` `INTERVAL` — temperamento equabile (default: 12 1200). - `INDEX`: divisioni dell’intervallo. - `INTERVAL`: ampiezza dell’intervallo, può essere intero/frazione (convertita in cents) o un numero di cents.

Opzioni aggiuntive: - `--no-reduce` — non ridurre i rapporti all’ottava/intervallo. - `--span` `N` (`--ambitus` `N`) — ripete la serie su `N` intervalli di ampiezza `INTERVAL` (default 1). - `--interval-zero` — forza `interval=0` nel `GEN -2` (usa `ratios` non ripetuti per la tabella `cpstun`). - `--export-tun` — esporta file AnaMark `.tun` basato su 128 note. - `--tun-integer` — per i file `.tun`: arrotonda i cents al valore intero più vicino (senza decimali). Default: due decimali.

Confronti: - `--compare-fund` <val> — fondamentale per 12-TET e armoniche nella tabella di confronto. Predefinito: `basenote` (la stessa usata per Hz custom). Accetta `basenote`, un nome nota o una frequenza. - `--compare-tet-align` {same, nearest} — allineamento 12-TET (default `same`) per calcolo del confronto stampato; in export viene derivato dal confronto log2 (round ai semitoni). - `--subharm-fund` <val> — fondamentale per subarmoniche (default `A5` se non specificato; internamente, se mancante, cade su `diapason`). Accetta nota o Hz. - `--midi-truncate` — in caso di overflow oltre 128 note, tronca la serie (altrimenti si cerca di adattare `basekey`).

Posizionale: - `OUTPUT_BASE` — nome base per i file di output (senza estensione).

# Sistemi di Accordatura

- 1) Naturale (4:5:6)
  - `--natural A_MAX B_MAX` produce combinazioni di quinte (3/2) e terze minori (5/4) potenziate secondo gli intervalli tra `-A_MAX..A_MAX` e `-B_MAX..B_MAX`, quindi normalizzate nell’ottava [1,2) salvo `--no-reduce`.
- 2) Danielou
  - `--danielou-all` genera una griglia estesa (53 valori unici se la riduzione all’ottava è attiva, con 2.0 come valore finale).
  - `--danielou a,b,c` consente di aggiungere manualmente rapporti:  $(6/5)^a * (3/2)^b * 2^c$ , poi ridotti all’ottava salvo `--no-reduce`. L’opzione è ripetibile.
  - Nota sugli esponenti negativi: racchiudere l’argomento tra virgolette per evitare ambiguità con le opzioni della shell/argparse (es.: `--danielou "-1,2,0"`).
- 3) Geometrico
  - `--geometric GEN STEPS INTERVAL` costruisce la sequenza  $r^0, r^1, \dots, r^{(STEPS-1)}$  dove  $r=GEN$ , ridotta nel dominio [1,INTERVAL) salvo `--no-reduce`.
  - INTERVAL**: intero senza suffisso = cents (es. 700 → 700 cents); per rapporto usa float o frazione (es. 2.0 o 2/1). Accetta anche cents con suffisso c (es. 700c).
- 4) Temperamento equabile (ET)
  - `--et INDEX INTERVAL` genera la radice  $ratio = \exp((INTERVAL/\ln(2))/1200)^{(1/INDEX)}$  e i rapporti  $ratio^i$  per  $i=0..INDEX-1$ . **INTERVAL** accetta frazione (convertita in cents) o numero di cents.

---

## File e Tabelle di Output

- 1) File Csound .csd
  - `OUTPUT_BASE.csd` contiene una riga `f` con `GEN -2: f <num> 0 <size> -2 numgrades interval basefreq basekey ratios...`
  - Il file viene creato con uno scheletro `<CsoundSynthesizer>` se assente; se esistente, viene aggiunta una nuova tabella con numero `f` incrementale.
  - La riga è preceduta da commenti allineati che indicano campi e metadati.
  - Se il `.csd` esisteva già, l’output base per gli altri file sarà `OUTPUT_BASE_<fnum>` (con `<fnum>` il numero della nuova tabella creata).
- 2) File .tun (opzionale, con `--export-tun`)
  - `OUTPUT_BASE.tun` con 128 righe `Note n=... cents` (cents assoluti riferiti a 8.1757989156437073336 Hz, riferimento Ana-Mark/A=440), ricavate dai `ratios` posizionati su `basekey` al di sopra della `basefrequency` (calcolata da `--basenote` e `--diapason`). Le note fuori dall’intervallo o sprovviste di ratio usano 12-TET rispetto alla base. Ordine: le righe sono per note MIDI 0→127; il segmento personalizzato derivato dai `ratios` è ordinato in modo crescente, così i cents aumentano con il numero di nota.
  - Formato numerico: per impostazione predefinita i valori sono arrotondati al più vicino con massimo due decimali; con `--tun-integer` i valori sono arrotondati ai cents interi, senza punto e senza decimali.
- 3) Tabelle del sistema
  - Testo: `OUTPUT_BASE_system.txt` con colonne: `Step, MIDI, Ratio, Hz`.
  - Excel: `OUTPUT_BASE_system.xlsx` (se `openpyxl` presente) con foglio “System”.
- 4) Tabelle di confronto
  - Testo: `OUTPUT_BASE_compare.txt` con colonne:
    - `Step, MIDI, Ratio, Custom_Hz, Harmonic_Hz, DeltaHz_Harm, Subharm_Hz, DeltaHz_Sub, TET_Hz, TET_Note, DeltaHz_TET`.
    - Dove `Custom_Hz` è la frequenza del sistema, `Harmonic_Hz` quella allineata della serie armonica, `Subharm_Hz` della subarmonica (filtrate nel range), `TET_*` derivati rispetto a `--compare-fund`.
  - Excel: `OUTPUT_BASE_compare.xlsx` (se `openpyxl` presente) con foglio “Compare”, intestazioni colorate e colonne di differenze assolute in Hz.
- 5) Stampa a video
  - Tabella multi-colonna “Step Hz” ottimizzata alla larghezza del terminale.
- 6) Intervallo per la tabella cpstun
  - Per default, se i `ratios` risultano nell’intervallo [1,2], `interval` è impostato a 2.0, altrimenti 0.0.
  - Con `--interval-zero` si forza `interval=0.0` nella riga `f` del `GEN -2` (in tal caso si usa la serie non ripetuta dai `ratios`).

---

# Gestione del Range MIDI

- La serie risultante, dopo un’eventuale estensione con `--span`, viene adattata con `basekey` per rientrare nel range 0..127.
  - Se `--midi-truncate` è attivo, eventuali eccedenze oltre 128 note vengono troncate partendo da `basekey`.
  - In assenza di truncation, il programma cerca di adattare `basekey` (se necessario) per includere tutti i passi e stampare una notifica.
- 

## Esempi di Utilizzo

1) 12-TET su ottava (default), base C4, diapason 440 Hz:

```
python SIM-2NV.py out
```

2) 19-TET sull’ottava (1200 cents):

```
python SIM-2NV.py --et 19 1200 out19
```

3) Sistema geometrico con generatore 3/2, 7 passi, intervallo 2/1:

```
python SIM-2NV.py --geometric 3/2 7 2/1 outGeom
```

4) Sistema geometrico con intervallo espresso in cents (quinta 700c):

```
python SIM-2NV.py --geometric 3/2 7 700c outGeomC
```

5) Intonazione naturale con A\_MAX=5, B\_MAX=4, senza riduzione all’ottava, span 2:

```
python SIM-2NV.py --natural 5 4 --no-reduce --span 2 outNat
```

6) Danielou (griglia completa):

```
python SIM-2NV.py --danielou-all outDan
```

7) Danielou manuale con più terne:

```
python SIM-2NV.py --danielou 1,0,0 --danielou 0,3,0 outDanM
```

8) Cambiare nota/frequenza base e generare .tun:

```
python SIM-2NV.py --basenote A4 --diapason 442 --export-tun outA442
```

8b) Esportare .tun con cents interi (arrotondamento al valore più vicino, senza decimali):

```
python SIM-2NV.py --export-tun --tun-integer out_int
```

9) Confronto: fondamentale di confronto diversa dalla base custom e subarmonica da 440 Hz:

```
python SIM-2NV.py --compare-fund A3 --subharm-fund 440 outCmp
```

10) Forzare interval=0 nella tabella cpstun, troncando per 128 note da basekey 36:

```
python SIM-2NV.py --interval-zero --basekey 36 --midi-truncate outZero
```

Note sugli output: se `out.csd` esiste già, la nuova tabella cpstun verrà aggiunta con un numero di funzione incrementale (es. `f 2`) e gli altri file useranno come base `out_2.*`.

---

## Note e Limitazioni

- Se `openpyxl` non è installato, le esportazioni `.xlsx` vengono saltate (restano i `.txt`).
- Intervallo/normalizzazione: `--no-reduce` disattiva la riduzione a [1,2) o [1,INTERVAL).
- Conversione nomi nota: sono accettati `C..B` con alterazioni `#` o `B` (bemolle) e numeri di ottava stile MIDI (C4=60). L’intervallo MIDI ammesso è 0..127.
- Serie armonica/subarmonica: sono calcolate entro limiti di sicurezza (`MAX_HARMONIC_HZ=10000`, `MIN_SUBHARMONIC_HZ=1`) e filtrate per allineamento alle frequenze custom ordinate.
- Differenze in Hz: le colonne `DeltaHz_*` esprimono lo scostamento assoluto o diretto rispetto a `Custom_Hz` per ogni confronto.

## Licenza

SIM version 1.75 Copyright (c) 2025 Luca Bimbi. Distrubuted under MIT License

MIT License

Copyright (c) 2025 Luca Bimbi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

## Crediti

- Programma: SIM — Sistemi di intonazione musicale.
- Questo manuale si riferisce specificamente a `SIM-2NV.py` (Versione 1.75).
- Copyright © 2025 Luca Bimbi. Licenza MIT.