

# An Algorithm to Compose Melody Lines.

## Part 2: Assigning Pitch

Cecilio Salmeron  
August, 2007

### Abstract

This document describes the algorithm used in the LenMus program to create single melody lines aimed at practicing music reading. The composition process takes place in two steps. The first step focuses at composing a rhythm line. This line must satisfy the constraints imposed by the rhythms that are going to be practised in the exercise. In the second step, pitch is assigned to the rhythm line in order to create a melody. This document describes the second step. The first step, composing the rhythm line, will be described in a future document.

The algorithm being described is based on the usual common practice guidelines for composing a melody for a given chord progression. The basic idea of the algorithm is to choose a harmonic progression and to assign beat notes to chord harmony notes. Notes not in beat position will be considered as non-chord notes. They are first classified, and then their pitch is assigned based on its type.

### Table of contents

- [1. Introduction](#)
- [2. The algorithm for pitch assignment](#)
  - [2.1 Choosing a harmonic progression](#)
  - [2.2 Contour curves](#)
  - [2.3 Classifying notes and assigning pitch to on-beat notes](#)
  - [2.4 Assigning pitch to non-chord notes](#)
- [3. The algorithm in detail](#)
- [4. Conclusions and future improvements](#)
- [Bibliography](#)

## 1. Introduction

When the LenMus program was started, there was a need for an automatic music composer to achieve a simple goal: to generate music lines to practice music reading (only solfege, excluding singing). Therefore the composer algorithm was focused on generating rhythm lines satisfying the constraints imposed by the exercise and the lesson. Pitch was assigned at random with only two restrictions:

1. It had to be within a predefined note range so that the resulting music line had all its notes meeting the lesson requirements.
2. The maximum interval between two consecutive notes was limited, so that the resulting score looked more 'real'. To improve the appearance, the first and last notes were forced to correspond to the root note of the tonality. Figure 1 shows a score generated by the described method.



*Figure 1 - Score generated by assigning random pitches to the notes.*

The scores produced in this way have no musical structure and, when played back, they give us the sensation of 'noise'. Notes are not harmonically related one to the other and the melody line does not follow any principle of connectivity: intervals between two consecutive notes are random intervals and the melody does not follow any line.

The scores generated in this way, although useful to practise rhythm reading, were not good enough to teach and learn

music. So it very soon became clear that the algorithm had to be reviewed in order to generate scores that resemble true melodic music lines. To this end, the approach chosen was to keep the rhythm line generation algorithm, since it works well, and to replace the random pitch assignation approach with a new algorithm. This algorithm should apply some of the music compositions rules of the common practise period. This idea seemed complex to implement and, since the goal was to generate music lines for practising music reading, it was decided to simplify the algorithm as much as possible and to program the minimal set of rules still able to produce useful scores.

To my surprise it was not very complex to implement some basic composition rules and the results were a notable improvement over the previous composer algorithm, as can be appreciated in figure 2. For better comparison, this score (and also the following ones) was generated using the same rhythm line used in figure 1.



Figure 2 - Score generated by the algorithm described in this document.

## 2. The algorithm for pitch assignment

The algorithm being described is based on the usual common practice guidelines for composing a melody to a given chord progression. The basic idea of the algorithm is to choose a harmonic progression and to assign beat notes to chord harmony notes. Notes not in beat position will be considered as melody embellishment, that is, non-harmony notes (also named non-chord notes) to "smooth out" melodic motion between or around chord tones. Their pitch is assigned in accordance with the non-chord notes composition rules. Therefore the algorithm is basically defined by three steps:

1. Choose a harmonic progression to base the melody line on and a 'contour curve' which is a shape that the notes of the melody line should follow approximately.
2. Assign pitch to the on-beat notes. Choose it from chord pitches.
3. Assign pitch to all other non-chord notes. The assigned pitch must satisfy the non-chord notes composition rules.

### 2.1. Choosing a harmonic progression

A harmonic progression (or chord progression) is a sequence of chords satisfying certain rules. Every chord corresponds to a harmonic function and limits the pitch of the chord notes to three tones.

In a first attempt to simplify the algorithm a chord is used for each measure. This way the number of chords to generate is just the number of measures in the score.

In this first version of the algorithm no attempt was made to synthesize a suitable chord progression satisfying the composition rules. Although developing a progression generator does not seem to be very difficult, it was considered more pragmatic to simply choose a progression from a predefined table of typical chord progressions.

Once a harmonic progression is chosen, we need to use tonality to compute the chord notes. In our case, tonality is a given parameter of the composer algorithm and is given by the desired key signature.

Two future improvements to the algorithm could be:

1. To consider more than one chord per measure depending on the time signature and the complexity of the rhythm line.
2. To include rules to synthesize the harmonic progressions instead of using a look up table with typical progressions.

### 2.2. Contour curves

Melody lines follow simple geometrical shapes, such as waving lines, arches or ramps. These line shapes define the 'contour line' of the melody. The idea is very well explained in [\[2\]](#).

In the described composer algorithm, contour curves are used to guide the process of assigning pitch to the notes. The algorithm selects, at random, a contour curve to be followed by the melody. In a first attempt I tried with arches and ramps. but after some experimentation arches were discarded because the generated music lines did not have a clear and sharp peak note. Triangle curves performed better and so all arch curves were replaced by triangles. In the current implementation the algorithm chooses at random between five possible basic shapes: triangle, ramp, zigzag, ramp+ triangle and

triangle+ramp (see figure 3). For each of these shapes two orientations are possible: up or down.

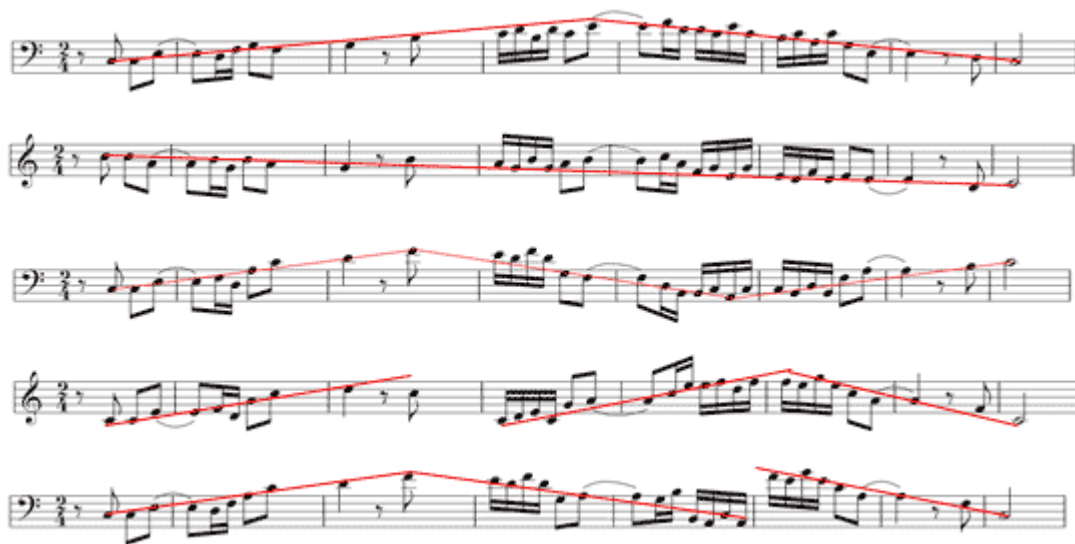


Figure 3 - Contour curves used in the algorithm.

After a shape is chosen, it is necessary to adjust the shape to the desired music score constraints (key signature and notes range). The curve has to satisfy the composition rule which states that the first and last notes of a melody should be the root note of the tonality. Therefore, the root note for the desired tonality is computed and start and end point curves are forced to be the root note.

### 2.3. Classifying the notes and assigning pitch to on-beat notes

The next step is to compute the points of the curve. As chord notes are assigned to on-beat notes, it is necessary to compute as many points as on-beat notes. So the score is analysed and the on-beat notes counted and identified, and once the needed number of points is known the algorithm proceeds to compute curve points (figure 4).



Figure 4 - A triangle curve computed for a melody with 16 beats, in C major with a notes range from A3 to G5. First (figure a), the rhythm line is used to identify and count the beat notes (12 notes). Next, (figure b) the same number of points are computed on the chosen contour curve. Notice that first and last points are adjusted to C pitch, as required by the key signature.

The next step is to assign pitch to the on-beat notes. This process just searches for the chord note that is nearest to the corresponding curve point (figure 5).

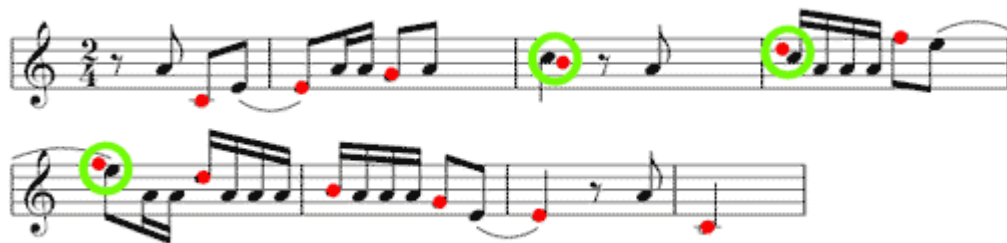


Figure 5 - The chord note that is nearest to the curve point is assigned to the corresponding on-beat note. In the example, chord progression I,V,I,IV,II,IV,I is used and points needing adjustment to match chord notes are marked with a circle. All other notes do not have any pitch assigned yet, but appear in the figure as A4 pitch.

## 2.4. Assigning pitch to non-chord notes

The last step is to assign pitch to the remaining notes. The strategy followed is to consider those notes that do not have a beat position as melody embellishment, that is, as non-harmony notes (also named non-chord notes) to "smooth out" the melodic motion between or around chord tones. A non-chord tone is a note which is not part of the accompanying harmony. For a clear on-line explanation of non-chord notes and their types see, for example, reference [3].

The process of assigning pitch to non-chord notes is based on determining how many notes there are between two consecutive on-chord notes, and on the interval that is formed by the on-chord notes. With this information the algorithm classifies the non-chord notes either as passing notes, as neighboring notes or as on-chord notes. For example, if the interval formed by the two on-chord notes is a unison and there is one non-chord note between them, the note is classified as 'neighboring tone'. Other non-chord note types, such as retardation, suspension, anticipation or appoggiatura, are not considered in current algorithm because their treatment is more complex.

Once the non-chord notes are classified, the last step is to assign pitch to the notes. This is a simpler process since it is just to follow the rules for the type of non-chord notes.



Figure 6 - The final score after assigning pitch to non-chord notes.

As a future improvement, more types of non-chord notes could be taken into account.

## 3. The algorithm in detail

The automatic music composer in full is programmed in file auxmusic/Composer6.cpp. The main line of the algorithm described in this document is in method InstantiateNotes(). The algorithm is as follows:

1. Choose a harmonic progression with as many chords as measures in the score. Compute the chord notes based on the key signature.
2. The contour curve will have as many points as on-chord notes in the music line. So this step is a loop to traverse the score and count the notes in beat position.
3. Choose at random a contour curve and compute its points. The curve will always start and finish in the root note. Its amplitude will be adjusted to satisfy the pitch constraints (min and max pitch).
4. This step is the main loop to assign pitch to the notes. The score is traversed again.
  1. For each chord note, assign the nearest pitch to the contour curve point that is on the progression chord assigned to current measure.
  2. For each pair of consecutive on-chord notes:
    1. Count how many non-chord notes there are in between them.
    2. Based on the interval between the on-chord notes and on the number of non-chord notes in between them, determine the more suitable non-chord notes group type (passing notes, neighboring notes, etc.).
    3. Assign pitch to the non-chord notes based on its type.
5. Done.

## 4. Conclusions and future improvements

The algorithm described is based on the usual common practice guidelines for composing a melody to a given chord progression. The basic idea of the algorithm is to choose a harmonic progression and to assign beat notes to chord harmony notes. Notes not in beat position are considered as non-chord notes. They are first classified, and their pitch is assigned based on its type.

The input to the algorithm is a rhythm line, a key signature and a pitch range. The algorithm adds musical structure to the rhythm line in its two musical dimensions:

1. The vertical dimension: harmony. The harmonic consistence is achieved by choosing a tonality and a harmonic progression following the traditional rules of composition,
2. The horizontal dimension: notes connectivity, that is, maintaining certain relationships between the intervals formed by consecutive notes. This is ensured by forcing the chord notes to follow a predefined 'contour curve', that is, the a shape for the melody line).

The algorithm is very simple and there is room for a lot of improvements. Three clear ones are 1) to consider more than one harmonic function per measure depending on the time signature and the complexity of the rhythm line; 2) to include rules to synthesize the harmonic progressions instead of using a look up table with typical progressions; and 3) to include more types of non-chord notes, not only passing tones and neighboring notes.

## Bibliography

- [1] Walter Piston, "Harmony", Lowe and Brydone, London, 1950
- [2] Robert J. Frank, Theory on the Web site, <http://www.smu.edu/totw/melody.htm>, 2000
- [3] Robert J. Frank, Theory on the Web site, <http://www.smu.edu/totw/nct.htm>, 2000