University of Massachusetts Amherst
ScholarWorks@UMass Amherst

Masters Theses 1911 - February 2014

2013

A Hierarchical Approach to the Analysis of Intermediary Structures Within the Modified Contour Reduction Algorithm

Kristen M. Wallentinsen
University of Massachusetts Amherst

Follow this and additional works at: https://scholarworks.umass.edu/theses

Part of the Music Theory Commons

Wallentinsen, Kristen M., "A Hierarchical Approach to the Analysis of Intermediary Structures Within the Modified Contour Reduction Algorithm" (2013). *Masters Theses 1911 - February 2014*. 1160.
Retrieved from https://scholarworks.umass.edu/theses/1160

This thesis is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses 1911 - February 2014 by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

A HIERARCHICAL APPROACH TO THE ANALYSIS OF INTERMEDIARY STRUCTURES WITHIN THE MODIFIED CONTOUR REDUCTION ALGORITHM

A Thesis Presented

by

KRISTEN M. WALLENTINSEN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF MUSIC

September 2013

Master of Music in Music Theory

© Copyright by Kristen M. Wallentinsen 2013

All Rights Reserved

A HIERARCHICAL APPROACH TO THE ANALYSIS OF INTERMEDIARY STRUCTURES WITHIN THE MODIFIED CONTOUR REDUCTION ALGORITHM

A Thesis Presented

by

KRISTEN M. WALLENTINSEN

Approved as to style and content by:

_____
Rob Schultz, Chair

_____
Gary Karpinski, Member

_____
Timothy Chenette, Member

_____
Jeffrey Cox, Department Chair,
Music and Dance

DEDICATION

To my mother and father, with love.

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Rob Schultz, for his patience and guidance throughout this entire process, as well as for inspiring my interest in contour theory. Without his constant support and valuable insight, this project would not have been possible. Thanks must also go to my other committee members, Dr. Gary S. Karpinski and Professor Timothy Chenette for their time and helpful feedback.

I must also thank the many friends who have supported me throughout this endeavor. I am indebted to David Mosher and Rebecca Long for listening to me ramble on about my ideas in the early stages, and to Corinne Salada for reading chapters and offering practical advice about the thesis process. Thanks also go to Sarah Weber, Renée Morgan, and Maki Matsui for their patient friendship and support these past few years.

I owe a tremendous debt of gratitude to my family for their endless love and encouragement. They have given me the strength and courage to pursue my dream, and without them I would be completely lost.

Finally, I would be remiss if I did not thank two very important educators, Gloria Velasco and Kathie Jarrett, who opened the doors to the world of music for me all those years ago.

ABSTRACT

A HIERARCHICAL APPROACH TO THE ANALYSIS OF INTERMEDIARY STRUCTURES WITHIN THE MODIFIED CONTOUR REDUCTION ALGORITHM

SEPTEMBER 2013

KRISTEN M. WALLENTINSEN, B.M., UNIVERSITY OF ARIZONA

M.M., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Rob Schultz

Robert Morris's (1993) Contour-Reduction Algorithm—later modified by Rob Schultz (2008) and hereafter referred to as the Modified Contour Reduction Algorithm (MCRA)—recursively prunes a contour down to its prime: its first, last, highest, and lowest contour pitches. The algorithm follows a series of steps in two stages. The first stage prunes c-pitches that are neither local high points (maxima) nor low points (minima). The second stage prunes pitches that are neither maxima within the max-list (pitches that were maxima in the first stage) nor minima within the min-list (pitches that were minima in the first stage). This second stage is repeated until no more pitches can be pruned. What remains is the contour's prime.

By examining how the reduction process is applied to a given c-seg, one can discern a hierarchy of levels that indicates new types of relationships between them. In this thesis, I aim to highlight relationships between c-segs by analyzing the distinct subsets created by the different levels obtained by the applying the MCRA. These subsets, or *sub-cseg*s, can be used to delineate further relationships between c-segs beyond their respective primes. As such, I posit a new method in which each sub-cseg produced by the MCRA is examined to create a system of hierarchical comparison that

measures relationships between c-segs, using sub-cseg equivalence to calculate an index value representing degrees of similarity. The similarity index compares the number of levels at which two c-segs are similar to the total number of comparable levels.

I then implement this analytical method by examining the similarities and differences between thirteen mode-2 Alleluias from the *Liber Usualis* that share the same alleluia and jubilus. The verses of these thirteen chants are highly similar in melodic content in that they all have the same prime, yet they are not fully identical. I will examine the verses of these chants using my method of comparison, analyzing intermediary sub-csegs between these 13 chants in order to reveal differences in the way the primes that govern their basic structures are composed out.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS v

ABSTRACT..... vi

LIST OF FIGURES x

1: INTRODUCTION 1

The Reductive Approach 8
Extensions to the Reductive Method 17

2: INTRODUCING THE HIERARCHICAL METHOD FOR COMPARING

CONTOURS WITH THE SAME PRIME AND THE SAME DEPTH LEVEL 19

Contour Reduction: The Recursive Approach.....20
Hierarchical Levels and Similarity Relations 23
Analytical Demonstration of the Hierarchical Comparison.....27
Wedge Shapes and the N=2 Problem.....31
Further Analytical Applications of This Comparative Theory 38
Conclusion 43

3: EXTENSIONS TO THE COMPARATIVE PROCESS 44

Primes on Differing Depth Levels 44
The Displacement Problem.....50
The Displacement Comparison.....52
The Similarity Index for Displaced C-segs.....57
An Application of the Comparative Process.....58
Conclusion 60

4: MEASURING SIMILARITY WITHIN ALLELUIAS OF THE SAME MODE 61

Applying the Hierarchical Comparison Method.....70
Contour and “Sameness” of Melody.....77

5: CONCLUSION.....116

Further Research.....118

APPENDIX: THE THIRTEEN COMMON MODE 2 ALLELUIAS.....121

REFERENCES.....129

LIST OF FIGURES

Figure Page

1.1. Application of Friedmann's CAS and CC 3

1.2. The COM-Matrix for the ⟨671254103⟩ c-seg 6

1.3. Comparison of COM-matrices for csegs ⟨043251⟩ and ⟨125403⟩ 7

1.4. Application of ACMEMB to c-segs of differing cardinalities (Marvin and Laprade 1987, 246). 9

1.5. Morris's contour reduction algorithm (Morris 1993, 212) 10

1.6. Application of Morris's contour reduction algorithm to the c-seg ⟨671254103⟩ 11

1.7. Illustration of the wedge-shape problem (Schultz 2008, 97). 12

1.8. The Modified Contour Reduction Algorithm (Schultz 2008, 108)..... 14

1.9. Application of the MCRA to the wedge-shaped c-seg 15

2.1. The Modified Contour Reduction Algorithm (Schultz 2008, 108)..... 21

2.2. Contour reduction and the sub-cseg..... 22

2.3. The comparison-hierarchy flow chart..... 24

2.4a. Score excerpt for measures 36–39 28

2.4b. Score excerpt for measures 114–116 28

2.4c. Comparison of c-segs for measures 38–39 and measures 114–115..... 29

2.5a. Score excerpt for measures 16 and 17..... 30

2.5b. C-seg comparison for measure 16 and measure 17 30

2.6. An illustration of the wedge-shape problem..... 32

2.7a. The Modified Contour Reduction Algorithm 35

2.7b. A correction of the wedge-shape problem 36

2.8a. The Alternate Modification of the Algorithm 37

2.8b. The alternate modification of the wedge-shape problem..... 38

2.9a. Score example of measures 1–9 40

2.9b. Comparison of c-segs for measures 1–4 and measures 5–6 41

2.10. Comparison of c-segs for measure 5 and measures 5–6 42

3.1a. Alleluia Cantate Domino..... 46

3.1b. Reduction of the verse from *Alleluia Cantate Domino* 47

3.1c. Reduction of the phrase "cantate domino"..... 48

3.1d. Reduction of the c-seg on the word "cantate" 49

3.2a. Phrases from Alleluia Dies Sanctificatus and Alleluia Hic Est Discipulus 49

3.2b. Reductions of the phrases presented in Figure 3.2a..... 50

3.3. Reduction of hypothetical c-segs to show displacement 51

3.4. Reductions of c-segs with ⟨021⟩ occurring at different depth levels 53

3.5. Displacement comparison flowchart..... 55

3.6. Comparison of displaced sub-csegs 58

4.1a. *Videntes stellam* (Robertson 28) 62

4.1b. A reduction of *Videntes stellam* to show the underlying arch structure 62

4.1c. Reductions of phrase 1 and phrase 2 of *Videntes stellam* 64

4.2a. *Alleluia Angelus Domini Descendit* 66

4.2b. The alleluia and jubilus from *Alleluia Angelus Domini Descendit*..... 67

4.2c. The verse from *Alleluia Angelus Domini Descendit*..... 68

4.2d. The reduction of the chant, *Alleluia Angelus Domini Descendit* 69

4.3. Ranges of chants in the eight modes from *Dialogus de musica* 71

4.4. A plagal chant wherein the full range of the plagal mode is not made apparent until the middle of the chant..... 71

4.5. Table of remaining primes seen in chant 73

4.6a. The reduction of *Alleluia Beatus vir* 74

4.6b. The reduction of *Alleluia Ego sum pastor bonus*..... 75

4.6c. The reduction of the alleluia and jubilus from *Alleluia Verumtamen* 76

4.6d. The reduction of the alleluia and jubilus from *Alleluia Propitus esto* 77

4.7a. The alleluia and jubilus of the 13 common mode-2 Alleluias 79

4.7b. Reduction of the alleluia and jubilus of the 13 common mode-2 Alleluias..... 79

4.8a. *Alleluia Dies sanctificatus*..... 80

4.8b. The reduction of Alleluia Dies sanctificatus..... 81

4.8c. Reduction of the verse from *Alleluia Dies sanctificatus*..... 82

4.8d. Reduction of the phrase *Dies sanctificatus illuxit nobis* from *Alleluia Dies sanctificatus* 83

4.8e. Reduction of the phrase *venite gentes et adorate dominum* from *Alleluia Dies sanctificatus* 84

4.8f. Reduction of the phrase *quia hodie Descendit lux magna super terram* from *Alleluia Dies sanctificatus*..... 85

4.8g. Prime sub-csegs and depth levels of *Alleluia Dies sanctificatus* 85

4.9a. *Alleluia Video caelos*..... 86

4.9b. Reduction of the chant *Alleluia Video caelos* 87

4.9c. Reduction of the verse from *Alleluia Video caelos*..... 88

4.9d. Reduction of the phrase *Video caelos apertos* from *Alleluia Video caelos* 89

4.9e. Reduction of the phrase *et Jesum stantem* from *Alleluia Video caelos* 90

4.9f. Reduction of the phrase *a dextris virtutis Dei* from *Alleluia Video caelos* 90

4.10. Comparison of *Alleluia Dies sanctificatus* and *Alleluia Video caelos* 91

4.11a. Similarity comparison of entire-chant c-segs with the chant *Alleluia Dies sanctificatus* 92

4.11b. Similarity comparison of verse c-segs with the chant *Alleluia Dies sanctificatus* 93

4.11c. Similarity comparison of first phrase c-segs with the chant *Alleluia Dies sanctificatus* 94

4.11d. Similarity comparison of second phrase chant c-segs with the chant *Alleluia Dies sanctificatus* 95

4.11e. Similarity comparison of third phrase chant c-segs with the chant *Alleluia Dies sanctificatus* 96

4.12. Formal framework for the 13 chants..... 99

4.13a. Normative model chant 100

4.13b. Reduction of normative model chant..... 101

4.13c. Reduction of the verse from the normative model chant 102

4.13d. Reduction of the first phrase from the normative model chant 103

4.13e. Reduction of the second phrase from the normative model chant..... 104

4.13f. Reduction of the third phrase from the normative model chant..... 104

4.14a. Similarity comparison of entire chant c-segs with the normative chant model..... 106

4.14b. Similarity comparison of verse c-segs with the normative chant model 107

4.14c. Similarity comparison of first phrase c-segs with the normative chant model..... 108

4.14d. Similarity comparison of second phrase chant c-segs with the normative chant model..... 109

4.14e. Similarity comparison of third phrase chant c-segs with the chant *Alleluia Dies sanctificatus* 110

4.15a Reduction of the phrase *Dies sanctificatus illuxit nobis* from *Alleluia Dies sanctificatus* 112

4.15b. Reduction of the phrase *Video caelos apertos* from *Alleluia Video caelos apertos* 113

A.1. *Alleluia Dies Sanctificatus* 121

A.2. *Alleluia Video Caelos* 122

A.3. *Alleluia Hic est discipulus* 122

A.4. *Alleluia Vidimus stellam* 123

A.5. *Alleluia Redemptionem* 123

A.6. *Alleluia Tu es Petrus* 124

A.7. *Alleluia Hic est sacerdos* 124

A.8. *Alleluia Sancti tui* 125

A.9. *Alleluia Magnus sanctus* 125

A.10. *Alleluia Nunc cum eo* 126

A.11. *Alleluia Inveni David* 126

A. 12. *Alleluia Tu puer Propheta* 127

A. 13. *Alleluia Domine diligo* 128

CHAPTER 1

INTRODUCTION

Early analytical approaches to contour treated it as only a highly generalized feature—i.e. a wave-like shape, or an arch-like shape. Arnold Schoenberg's *Fundamentals of Musical Composition* simply described contour as a "feature" of a motive—that which makes a motive distinctive—with no further definition of how to discuss contour, or how to use it as a tool of composition (1967, 9). Other cases include Peter Wagner's description of chant: "as a rule the melodic line begins at a low pitch, rises to a point of climax and gradually descends to its final," (1911, III.9; quoted in Stevens 1986, 279-80) again describing only a general arch-shape motion of the melody. Ernst Toch likewise approached a discussion of melodic shape in general terms, stating that "with the combination of ascending and descending scale-segments melody approaches its real nature: the *wave line*" (1948, 78). Toch also discussed melodic lines as constituting several small waves adding up to one large wave (1948, 79–80). Despite this further attempt at describing melodic structure, none of these discussions of melodic shape did anything to systematize the analysis of contour to the degree that other aspects of music—such as rhythm, harmony, or counterpoint—have been. Instead, such highly metaphorical descriptions served a subordinate role, only reinforcing points made about other topics.

First attempts to systematically account for the content of a melodic contour came when ethnomusicologists such as Charles Seeger and Charles Adams attempted to use contour as a method of categorizing melodies of the groups they were studying. Seeger

(1960) addressed pitch direction as a varied function of music, referring to a rise in pitch as "tension" and indicating it with a plus sign (+). Likewise, Seeger referred to a fall in pitch as "detension," indicated by a minus sign (–), and to a maintenance of pitch as "tonicity," indicated by an equal sign (=). He used these symbols to categorize patterns of tension, detension, and tonicity as they related to pitch direction, among other musical parameters.[1]

Adams (1976) is the first to strictly define melodic contour. He states that contour is "the product of distinctive relationships among the minimal boundaries of a melodic segment" (Adams 1976, 195). He defines such minimal boundaries as

> those pitches which are considered necessary and sufficient to delineate a melodic segment, with respect to its temporal aspect (beginning-end) and its tonal aspect (tonal range). Bounding a series of pitches by an initial pitch (I), a final pitch (F), a highest pitch (H), and a lowest pitch (L), satisfies these conditions while defining fewer or more boundaries does not (196).

These four boundary pitches became an integral part of later contour theories, including Morris's reductive approach.

Despite its origins, contour theory found a more permanent home in the analysis of post-tonal music. With post-tonal music, the need to categorize melodic segments by means other than the tonal syntax that came with common practice tonality arose, and because of this, melodic contour began to serve a much more independent role in subsequent analyses. Building on the earlier work of Adams, five influential authors have continued to develop techniques for studying contour within musical analysis:

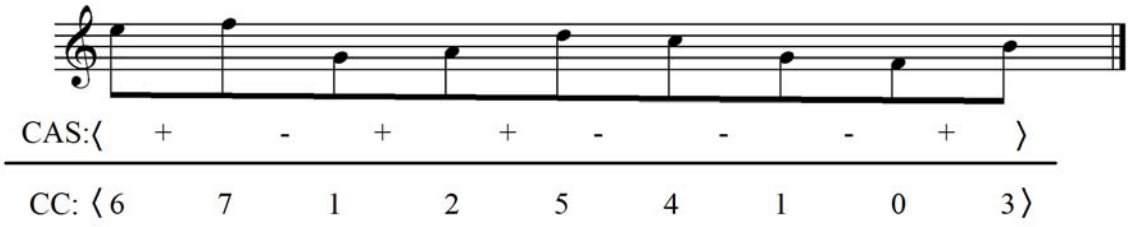
---

[1] Seeger also used these terms to refer to changes in dynamics and tempo, stating that the type of direction change (i.e., louder vs. softer or faster vs. slower) fell into the same sort of binary continuum as pitch rising or falling. In this way, he attempted to unify these musical elements under one system of description.

Michael Friedmann (1985), Robert Morris (1987), and Elizabeth West Marvin and Paul Laprade (1987).

Friedmann categorizes melodic contours based on two concepts: the relationships between adjacent pitches and relative positions between all pitches in a contour segment, or c-seg. Relationships between adjacent pitches provide detailed information about the immediate surface-level structure of a c-seg. His *Contour Adjacency Series* (CAS) outlines a sequential series of direction changes as the means of defining the contour, using (+) and (–) symbols to account for such changes.[2] For example, the contour of the passage shown in Figure 1.1 would have a CAS of ⟨+,–,+,+,–,–,–,+⟩. Friedmann then constructs vectors for the CAS by tallying the ascents and descents in the CAS. The CAS vector of the c-seg in Figure 1.1 would be ⟨4,4⟩ indicating an equal number of ups and downs in the c-seg. The CAS is a good tool for describing the note-to-note contour features of a c-seg, but this approach does not account for the global contour properties that give a c-seg its distinct shape.

Figure 1.1. Application of Friedmann's CAS and CC



[2] Friedman derives the (+) and (–) symbols from John Rahn's ordered pitch intervals, simply removing the intervallic distance factor.

To address this, Friedmann created the *Contour Class* (CC). Instead of looking only at adjacent pitches, the CC examines the relative positions between all pitches within a c-seg. In the CC, each unique pitch is given an ordinal number, with 0 representing the lowest pitch, and n–1 representing the highest pitch within a set of n pitches. This gives a global view of all the pitches in the set with regard to their registral position. Returning to the c-seg from Figure 1.1, the contour of the opening measure has a CC of ⟨6–7–1–2–5–4–1–0–3⟩. A CC vector counting the ups and downs in a similar manner to a CAS vector now takes into account relationships between non-adjacent pitches. Using the CC, Friedmann constructs a *Contour Interval Array* (CIA), which functions in a manner similar to an interval vector in set theory. The CIA takes the relative distances between each pitch in the contour and accounts for the "multiplicity of each contour interval type in the CC as a whole" (Friedmann 1985, 230). Given a contour with a CC of ⟨0–4–1–2–5–3⟩, the CIA would be ⟨4,2,2,2,1/1,2,1,0,0⟩.

With the CC and the CIA in hand, Friedmann creates two types of vectors for the contour class. *Contour class vector* I (CCV I) presents a "two-digit summation of the degrees of ascent and descent expressed in a CIA. The first digit is the total of the products of the frequency and contour interval type found on the left side of the slash in the middle of the CIA" (1985, 247). The second digit follows the same logic as the first, but on the right side of the slash, representing the descents. For the CIA expressed above, the first digit of CCV I would be 4(1) + 2(2) + 2(3) + 2(4) + 1(5), or 27. The second digit would be 1(1) + 2(2) + 1(3) + 0(4) + 0(5), or 8. Therefore the CCV I is ⟨27,8⟩. *Contour class vector* II (CCV II) is a bit more general than CCV I, discarding contour interval size and examining ascent vs. descent in a similar manner to the CAS

vector, only now including non-adjacent pitches in the calculation. The CCV II counts the total number of ascents and descents represented in the CC by adding the numbers on either side of the CIA. Therefore the CC II for the CIA mentioned above (⟨4,2,2,2,1/1,2,1,0,0⟩) would be ⟨11,4⟩. Both the CC and the CAS allow for classification of contours based on equivalences of contours bearing identical values for these measurements.

In another influential discussion of contour, Morris (1987) introduces specific methodology for contour description. He begins by defining contour space (c-space): "a c-space of order n, is a pitch-space of n elements, called c-pitches (cps). C-pitches are numbered in order from low to high, beginning with 0 up to *n*–1. The intervallic distance between the cps is ignored and left undefined" (26).[3] Morris also puts forth a method for comparing all c-pitches within a c-seg with each other. His *comparison matrix* (COM-matrix) compares every ordered pair of c-pitches in the c-seg. For example, the COM-matrix for the ⟨671254103⟩ c-seg discussed above, shown in Figure 1.2, provides a comprehensive array of the relative position of each c-pitch within the c-space. The first row in the matrix compares all c-pitches against the first c-pitch in the c-seg, the 6. The second row compares all c-pitches against the 7, and so forth for each c-pitch in the c-seg. In this way, it not only compares adjacent c-pitches, but all non-adjacent c-pitches as well.

[3] Morris's description of c-pitch numbering is equivalent to Friedman's CC.

Figure 1.2. The COM-Matrix for the ⟨671254103⟩ c-seg

	6	**7**	**1**	**2**	**5**	**4**	**1**	**0**	**3**
6	0	+	–	–	–	–	–	–	–
7	–	0	–	–	–	–	–	–	–
1	+	+	0	+	+	+	0	–	+
2	+	+	–	0	+	+	–	–	+
5	+	+	–	–	0	–	–	–	–
4	+	+	–	–	+	0	–	–	–
1	+	+	0	+	+	+	0	–	+
0	+	+	+	+	+	+	+	0	+
3	+	+	–	–	+	+	–	–	0

Marvin and Laprade (1987) develop two additional ways to compare contours. They use the COM-matrix in order to illuminate similarities between both adjacent and non-adjacent c-pitches. The first method, called the *contour similarity function* or CSIM, uses the COM-matrix to compare ascents and descents among both adjacent and non-adjacent cpitches of two c-segs in order to arrive at a ratio between identical motions and different motions. To calculate CSIM, one only needs to compare corresponding positions within the upper right-hand triangles of the matrices.[4] The number of identical positions within the COM-matrix is then divided by the total number of positions compared in order to arrive at the CSIM value. Figure 1.3 illustrates this principle: COM-matrices are shown for c-segs ⟨043251⟩ and ⟨125403⟩. Each position in the upper right-hand triangle of the matrix is compared to reveal that there are 7 identical positions, out of a total of 15 positions compared. Therefore the CSIM value for these two c-segs is 7/15, or 0.47.

CSIM is an effective method for arriving at a quantifiable similarity measurement between two c-segs. However, it does have one fundamental restriction: it allows for the

[4] It is necessary to examine only the upper right-hand triangle of the matrices because "the comparison matrix always displays a symmetry of inverse signs around the main diagonal" (Morris 1987, 28).

comparison of c-segs of the same cardinality only—that is, c-segs with the same number of c- pitches. Marvin and Laprade therefore devise a second method for comparing c-segs: the *contour embedding function* or CEMB. Instead of measuring the number of similar movements between c-pitches, CEMB measures how many occurrences of the smaller c-seg are embedded within the larger c-seg, and compares that value against the total number of csubsegs with the same cardinality as the smaller c-seg. For example, a c-seg of ⟨021⟩ is embedded within the c-seg ⟨023154⟩ seven times: the ⟨021⟩, ⟨031⟩, ⟨054⟩, ⟨231⟩, ⟨254⟩, ⟨354⟩, and ⟨154⟩ all become ⟨021⟩ under Marvin and Laprade's *translation* operation—that is, the renumbering of csubsegs accordingly in register from 0 to *n*–1, where *n* now represents the cardinality of the csubseg. There are 20 possible cardinality-3 csubsegs within the ⟨023154⟩ c-seg, so the CEMB function would return a value of 7/20, or 0.35.

Figure 1.3. Comparison of COM-matrices for csegs ⟨043251⟩ and ⟨125403⟩

c-seg A: ⟨043251⟩

	0	4	3	2	5	1
0	0	⊕	⊕	⊕	+	⊕
4	–	0	–	–	+	–
3	–	+	0	⊖	+	⊖
2	–	+	+	0	+	⊖
5	–	–	–	–	0	–
1	–	+	+	+	+	0

c-seg B: ⟨125403⟩

	1	2	5	4	0	3
1	0	⊕	⊕	⊕	–	⊕
2	–	0	+	+	–	+
5	–	–	0	⊖	–	⊖
4	–	–	+	0	–	⊖
0	+	+	+	+	0	+
3	–	–	+	+	–	0

CSIM (A,B) = 7/15 = 0.47

Marvin and Laprade make additional refinements to the CEMB function, introducing mutually embedded csubsegs in order to provide a more complete picture of the relationships within two c-segs. The ACMEMB function counts the number of identical mutually embedded csubsegs within the two c-segs in question. An example of ACMEMB is shown in Figure 1.4. The figure shows that there are 33 mutually embedded csubsegs within the two c-segs in question, out of a total of 37 possible csubsegs.

The Reductive Approach

Morris (1993) introduced the notion of perceptual hierarchy in musical contour theory with his Contour-Reduction Algorithm. This algorithm, shown in Figure 1.5, follows a series of steps in two stages. The first stage prunes c-pitches that are neither local low points (minima) nor high points (maxima). The second stage prunes pitches that are neither maxima within the max-list (the collection of pitches that were maxima in the first stage) nor minima within the min-list (the collection of pitches that were minima in the first stage). This second stage is repeated as many times as is necessary until no more pitches can be pruned. What remains is the contour's *prime*, or its initial, final, highest, and lowest c-pitches. To illustrate this concept, Figure 1.6 applies the steps in Morris's algorithm to the c-seg from Figure 1.1, ⟨671254103⟩. The algorithm begins by deleting pitches that are not local minima or maxima, arriving at the c-seg ⟨451302⟩. Entering stage two of the algorithm at N=1 on line B, steps 6 and 7 flag the maxima in the max-list (c-pitches 4, 5, and 2) and the minima in the min-list (c-pitches 4,

Figure 1.4. Application of ACMEMB to c-segs of differing cardinalities (Marvin and Laprade 1987, 246).[5]

C-seg A: 〈0213〉	**C-seg B: 〈02134〉**
Csubsegs	
〈02〉 = 〈01〉	〈02〉 = 〈01〉
〈01〉 = 〈01〉	〈01〉 = 〈01〉
〈03〉 = 〈01〉	〈03〉 = 〈01〉
〈21〉 = 〈10〉	〈04〉 = 〈01〉
〈23〉 = 〈01〉	〈21〉 = 〈10〉
〈13〉 = 〈01〉	〈23〉 = 〈01〉
〈021〉 = 〈021〉	〈24〉 = 〈01〉
〈023〉 = 〈012〉	〈13〉 = 〈01〉
〈013〉 = 〈012〉	〈14〉 = 〈01〉
〈213〉 = 〈102〉	〈34〉 = 〈01〉
〈0213〉 = 〈0213〉	〈021〉 = 〈021〉
	〈023〉 = 〈012〉
	〈024〉 = 〈012〉
	〈013〉 = 〈012〉
	〈014〉 = 〈012〉
	〈034〉 = 〈012〉
	〈213〉 = 〈102〉
	〈214〉 = 〈102〉
	〈234〉 = 〈012〉
	〈134〉 = 〈012〉
	〈0213〉 = 〈0213〉
	〈0214〉 = 〈0213〉
	〈0234〉 = 〈0123〉
	〈0134〉 = 〈0123〉
	〈2134〉 = 〈1023〉
	〈02134〉 = 〈02134〉
33 csegs mutually embedded in csegs A and B: ACMEMB (A,B) = 33/37 = 0.89	

[5] The c-segs in this example come from Marvin and Laprade's discussion of ACMEMB. However, these two c-segs were originally c-seg B and c-seg C in their example.

Figure 1.5. Morris's contour reduction algorithm (Morris 1993, 212)

Definition: *Maximum pitch:* Given three adjacent pitches in a contour, if the second is higher than or equal to the others it is a maximum. A set of maximum pitches is called a maxima. The first and last pitches of a contour are maxima by definition.

Definition: *Minimum pitch*: Given three adjacent pitches in a contour, if the second is lower than or equal to the others it is a minimum. A set of minimum pitches is called a minima. The first and last pitches of a contour are minima by definition.

Algorithm: Given a contour C and a variable N:

[STAGE ONE:]

Step 0: Set N to 0.

Step 1: Flag all maxima in C; call the resulting set the *max-list*.

Step 2: Flag all minima in C; call the resulting set the *min-list*.

Step 3: If all pitches in C are flagged, go to step 9.

Step 4: Delete all non-flagged pitches in C.

Step 5: N is incremented by 1 (i.e., N becomes N + 1).

[STAGE TWO:]

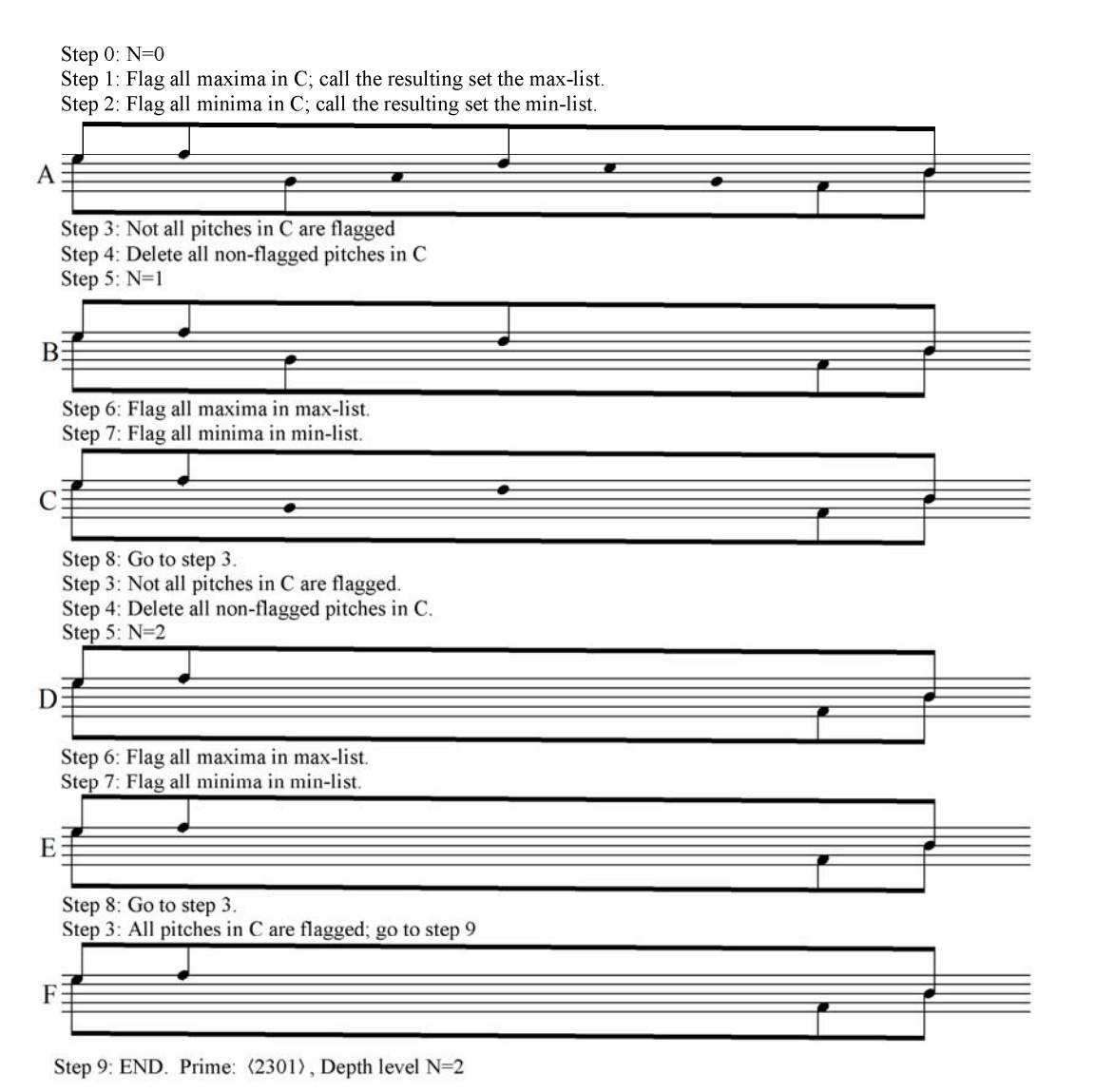
Step 6: Flag all maxima in max-list. For any string of equal and adjacent maxima in max-list, either: (1) flag only one of them; or (2) if one pitch in the string is the first or last pitch of C, flag only it; or (3) if both the first and last pitch of C are in the string, flag (only) both the first and last pitch of C.

Step 7: Flag all minima in min-list. For any string of equal and adjacent minima in min-list, either: (1) flag only one of them; or (2) if one pitch in the string is the first or last pitch of C, flag only it; or (3) if both the first and last pitch of C are in the string, flag (only) both the first and last pitch of C.

Step 8: Go to step 3.

Step 9: End. N is the "depth" of the original contour C.

Figure 1.6. Application of Morris's contour reduction algorithm to the c-seg ⟨671254103⟩

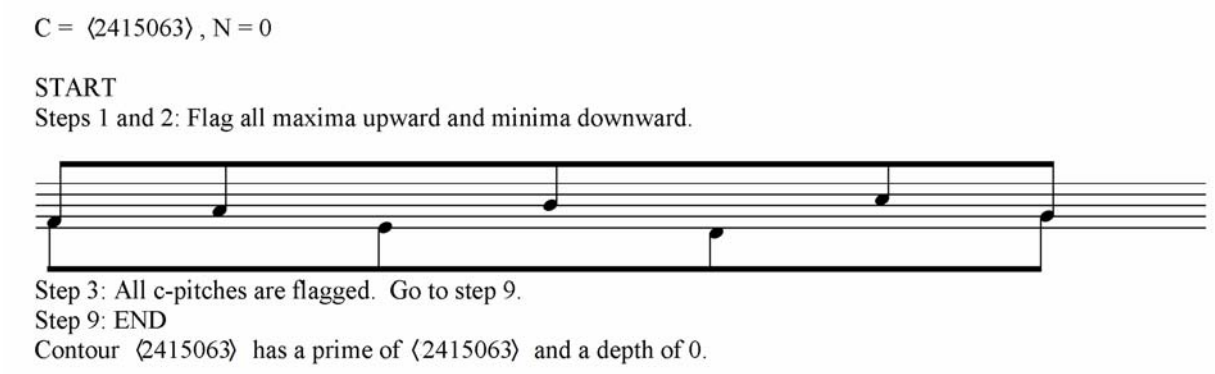


0, and 2). The algorithm then proceeds back to step 3, where the non-flagged pitches are deleted, leaving ⟨2301⟩ at N=2 on line D. Proceeding through steps 6 and 7 a second time, we find no more pitches to delete. Since all pitches are flagged, the reduction ends, yielding a prime of ⟨2301⟩ and a depth level of N=2.

Although the contour reduction algorithm outlined by Morris is useful in providing a structured approach to finding the prime of a c-seg, Morris leaves the algorithm open-ended to allow for different modifications as the need arises. For example, it allows for variability in the pruning of repeated c-pitches, and it does not allow for stage-two pruning of c-segs that do not get pruned in stage one of the algorithm.

Rob Schultz (2008) discusses c-segs that feature "a progressive outward expansion of c-pitches in c-space, thereby forming a wedge shape" (96). His example, reproduced in Figure 1.7, shows that Morris's algorithm would call the c-seg ⟨2415063⟩ a prime without proceeding on to stage two, since all c-pitches in the c-seg would be flagged as either a maxima or minima. This becomes a problem when one consults the list of primes Morris provides in his article, only to find that ⟨2415063⟩ is not on the list. Schultz posits that each stage of the algorithm must be applied to every c-seg "at least once in order to reliably produce a true prime" (Schultz 2008, 96).

Figure 1.7. Illustration of the wedge-shape problem (Schultz 2008, 97).



Schultz points out one other problem with the reduction algorithm. In a c-seg such as ⟨2414043⟩, step 6 of the algorithm states that only one of the 4s is to be retained, yet the decision to flag any one of the 4s at the expense of the other two can change the resulting prime of the reduction.

To account for these loopholes, Schultz introduces several modifications to Morris's algorithm. Figure 1.8 shows Schultz's modified contour reduction algorithm, which I will refer to as the *MCRA*. Schultz adds steps 8–12, and modifies steps 3, 6, and 7, in order to account for the specific problems discussed above. Step 3 now directs the reduction to stage two of the algorithm in the event that all c-pitches are flagged, thereby addressing the problem of wedge-shaped c-segs. Steps 6 and 7 now flag all c-pitches in a string of equal and adjacent maxima or minima (unless certain criteria are met), putting off the pruning of repetitions until the next steps. Steps 8 and 9 remove extraneous flags from any string of maxima in the max-list for which no minima intervene. Steps 10, 11, and 12 then remove all pairs of repetitions (i.e. a series of three or more equal and adjacent maxima wherein equal and adjacent minima intervene) in the max- and min-lists except for the outermost c-pitches involved in the repetition. These added steps address the problem of pitch repetition in the max- or min-lists (Schultz 2008, 106). Under the MCRA, the problem highlighted by Figure 1.7 has been corrected. Figure 1.9 displays the corrections to the reductive process for these c-segs: now the c-seg in Figure 1.9 reduces beyond stage one.

Figure 1.8. The Modified Contour Reduction Algorithm (Schultz 2008, 108)
Algorithm: Given a contour C and a variable N:

Step 0: Set N to 0

Step 1: Flag all maxima in C upwards; call the resulting set the *max-list*

Step 2: Flag all minima in C downwards; call the resulting set the *min-list*

Step 3: If all c-pitches are flagged, go to step six

Step 4: Delete all non-flagged c-pitches in C

Step 5: N is incremented by 1 (i.e., N becomes N+1)

Step 6: Flag all maxima in the max-list upwards. For any string of equal and adjacent maxima in the max-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 7: Flag all minima in the min-list downwards. For any string of equal and adjacent minima in the min-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 8: For any string of equal and adjacent maxima in the max-list in which no minima intervene, remove the flag from all but (any) one c-pitch in the string.

Step 9: For any string of equal and adjacent minima in the min-list in which no maxima intervene, remove the flag from all but (any) one c-pitch in the string.

Step 10: If all c-pitches are flagged, and no more than one c-pitch repetition in the max-list and min-list (combined) exists, not including the first and last c-pitches of C, proceed directly to step 17.

Step 11: If more than one c-pitch repetition in the max-list and/or min-list (combined) exists, not including the first and last c-pitches of C, remove the flags on all repeated c-pitches except those closest to the first and last c-pitches of C.

Step 12: If both flagged c-pitches remaining from step 11 are members of the max-list, flag any one (and only one) former member of the min-list whose flag was removed in step 11; if both c-pitches are members of the min-list, flag any one (and only one) former member of the max-list whose flag was removed in step 11.

Step 13: Delete all non-flagged c-pitches in C

Step 14: If $N \neq 0$, N is incremented by 1 (i.e., N becomes N+1)

Step 15: if $N = 0$, N is incremented by 2 (i.e., N becomes N+2)

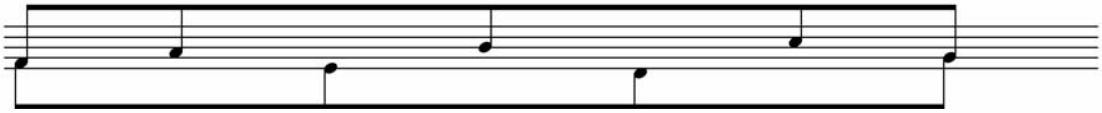
Step 16: Go to step 6

Step 17: End. N is the "depth" of the original contour C.

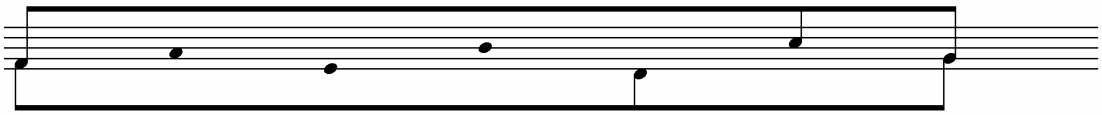
The reduced contour is the prime of C; if N=0, then the original C has not been reduced and is a prime itself.

Figure 1.9. Application of the MCRA to the wedge-shaped c-seg

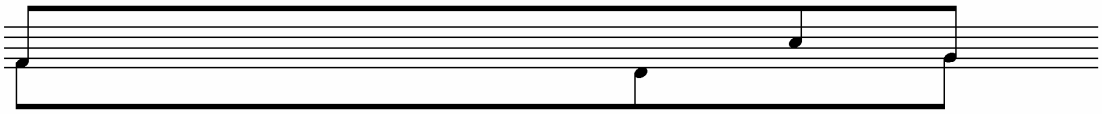
C= ⟨2415063⟩ , N=0
Steps 1 and 2: Flag all maxima upward and minima downward.



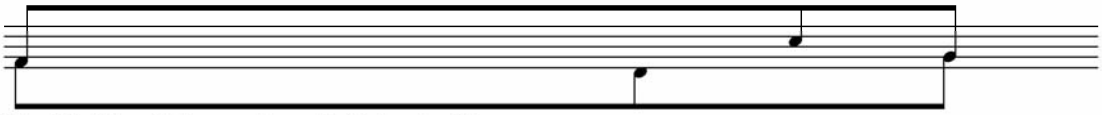
Step 3: All c-pitches are flagged. Go to step 6
Steps 6 and 7: Flag all maxima in the max-list and minima in the min-list.



Steps 8 and 9: Not applicable
Step 10: Not all c-pitches are flagged
Steps 11 and 12: Not applicable
Step 13: Delete non-flagged c-pitches



Step 15: N=2
Step 16: Go to steps 6 and 7: Flag all maxima in the max-list and minima in the min-list
Steps 8 and 9: Not applicable



Step 10: All c-pitches are flagged. Go to step 17.
Step 17: END
Prime: ⟨1032⟩, Depth level 2

Morris uses his algorithm to arrive at a table of basic prime classes that occur in music.

> Each class is a cseg class and thus includes primes related by R [retrograde], I [inversion], and RI [retrograde inversion]. There are 25 basic prime classes and 28 secondary prime classes—53 classes in all, only five of which do not have repetitions or simultaneities. The five are ⟨0⟩, ⟨01⟩, ⟨021⟩, ⟨1032⟩, and ⟨1302⟩. These together with ⟨010⟩ and ⟨1021⟩ are called the *linear prime classes*. (Morris 1993, 218)

These linear prime classes, to which Schultz added ⟨10201⟩ and ⟨10302⟩, provide a systematic organizational scheme with which to illustrate relationships between c-segs.

Such relationships between primes and their respective prime classes form the basis of most comparisons made using the MCRA.

Building from Morris's algorithm from a different angle, Mustafa Bor (2009) introduces a similar algorithm in which contours are reduced using what he calls "window algorithms." The window algorithm essentially isolates specific subsets within a c-seg and looks for pitches that are neither local high and low points within the window to prune. His 3-window, for example, prunes a medial pitch within a group of three pitches if it is neither a minimum nor a maximum. The window is designed to move forward linearly, as if in time, in order to treat each pitch in turn as a medial pitch. It is this 3-window approach that is in operation in the successive pruning of the MCRA. One can also have a 5-window that follows a similar model of pruning, as well as a 7-window, a 9-window, and so on. Such window algorithms follow similar principles, but on a larger scale. The 5-window, for example, expands the window to 5 pitches, and once again evaluates the pitch in the middle of the window, pruning if it is neither a maximum nor a minimum. Through the 5-window, Bor is able to examine non-adjacent pitches, determining maxima and minima that would be pruned in stage two of Morris's algorithm. Like Morris's algorithm, Bor's 3-window algorithm always prunes pitches so that the contour becomes a series of alternating signs with no passing pitches intervening, while the 5-window algorithm reduces the contour further to arrive at an irreducible result. Despite these similarities, Bor's window algorithms differ in that they use only one method of reduction: the window pruning method. In this method, maxima are never compared directly with one another, nor are minima, producing a different approach to the reductive process. It is hypothetically possible to have a contour that possesses

minima (under the Morris sense of the term) that are higher in pitch than other maxima pitches.

Extensions to the Reductive Method

This thesis posits a new method of analysis using a modified version of the MCRA presented above, wherein each stage of the algorithm is taken into account to create a system of hierarchical levels that we can use to measure relationships between c-segs. I aim to highlight relationships between c-segs by analyzing the distinct subsets created by the recursions at different levels within the algorithm. I also use this method of comparison to calculate an index value representing degrees of similarity between two c-segs. This similarity index compares the number of levels deemed equivalent to the total number of comparable levels within the c-segs.

To illustrate the analytical approaches of this hierarchical method of comparison, I will make a case for the use of the hierarchy in the analysis of plainchant, in order to study how techniques of melodic composition manifest themselves in contour similarity. I will explore contour's interaction with modal tendencies, and I will use the MCRA and comparative process to more accurately define the intricate shapes of melismatic chant structure. Specifically, I will illustrate the usefulness of this analytical method by discussing the similarity and difference between thirteen mode-2 *Alleluia* chants from the *Liber Usualis* that share the same *alleluia* and jubilus. These thirteen chants share very similar verses, yet the melodic content of these verses is not identical. I will examine the verses of these chants using my method of comparison, in order to determine the overall similarity of these chants from a contour perspective. I will also use the method of

comparison outlined above to highlight the differences between the verses at levels shallower than prime, and to explore possible causes for these differences.

CHAPTER 2

INTRODUCING THE HIERARCHICAL METHOD FOR COMPARING CONTOURS WITH THE SAME PRIME AND THE SAME DEPTH LEVEL

Using the MCRA, this chapter explores the significance of similarity between c-segs with the same prime. It modifies the reduction algorithm further to allow for direct correspondence among depth levels in order to determine a degree of similarity between two c-segs. It also discusses levels shallower than prime, introducing a method of determining similarity using these levels as a basis, and explore possible applications of this comparative theory.

The goal of the algorithm, as Morris originally conceived of it, was to provide a rigorously structured approach to arriving at the salient boundary pitches of a contour—the first, last, highest, and lowest. The recursions within the algorithm give us the prime contour at the end, and also produce a hierarchy of levels above that prime. Morris even alludes to the importance of these levels: he states that "each stage of reduction provides a contour on a distinct analytic *level*" (1993, 213) and draws a comparison to Schenkerian hierarchical levels of structure (215). Despite this initial nod to the importance of these intermediary levels between the prime and the surface, much of Morris's discussion focuses on equivalence based solely on prime contours and their resulting depth levels. However, examining contour reductions at specific depth levels other than the prime of a contour segment (c-seg) provides for a more nuanced comparison.

Contour Reduction: The Recursive Approach

The MCRA consists of 17 steps, as seen in Figure 2.1. Morris states that a c-seg of cardinality *n* can be reduced to a smaller "prime" using these steps. The algorithm follows these steps in what Bor (2009) called two stages. The first stage includes steps 0 through 5, where the c-seg is pruned of c-pitches that were neither local minima nor maxima. The second stage includes steps 6 through 17, and now prunes pitches that are not maxima within the max-list (the collection of pitches that were maxima in the first stage) nor minima within the min-list (the collection of pitches that were minima in the first stage).[6] This second stage is then repeated as many times as necessary until no more pitches can be pruned.

Figure 2.2 provides an illustration of the application of the MCRA. A c-seg of cardinality 7 (with pitches numbered 0 to *n*-1), ⟨1312014⟩, is displayed on a clefless, five-line staff, which is used to represent contour pitches in contour space, as opposed to pitches in pitch space. The staves themselves are labeled in alphabetical order for ease of reference. Steps 1–4 are applied to the surface-level c-seg, pruning all but the maxima and minima in the first stage. At this point, we arrive at our first *sub-cseg*, ⟨131204⟩, or $sub\text{-}cseg_1$ as shown on staff B.[7] This sub-cseg represents the first level deeper than the

[6] Several of the steps in stage two apply to specific conditions that occur only in certain c-segs. Specifically, steps 8-9 and 11-12 deal with the eventuality of repeated pitches in the max- and min-lists, and are not necessarily used in c-segs that do not display such repetitions.

[7] Marvin and Laprade have a similar term, the csubseg, which refers to any contiguous or non-contiguous subset within a c-seg. I am using the term sub-cseg here to refer to very specific non-contiguous subsets produced by the algorithm.

Figure 2.1. The Modified Contour Reduction Algorithm (Schultz 2008, 108)
Algorithm: Given a contour C and a variable N:

Step 0: Set N to 0

Step 1: Flag all maxima in C upwards; call the resulting set the *max-list*

Step 2: Flag all minima in C downwards; call the resulting set the *min-list*

Step 3: If all c-pitches are flagged, go to step six

Step 4: Delete all non-flagged c-pitches in C

Step 5: N is incremented by 1 (i.e., N becomes N+1)

Step 6: Flag all maxima in the max-list upwards. For any string of equal and adjacent maxima in the max-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 7: Flag all minima in the min-list downwards. For any string of equal and adjacent minima in the min-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 8: For any string of equal and adjacent maxima in the max-list in which no minima intervene, remove the flag from all but (any) one c-pitch in the string.

Step 9: For any string of equal and adjacent minima in the min-list in which no maxima intervene, remove the flag from all but (any) one c-pitch in the string.

Step 10: If all c-pitches are flagged, and no more than one c-pitch repetition in the max-list and min-list (combined) exists, not including the first and last c-pitches of C, proceed directly to step 17.

Step 11: If more than one c-pitch repetition in the max-list and/or min-list (combined) exists, not including the first and last c-pitches of C, remove the flags on all repeated c-pitches except those closest to the first and last c-pitches of C.

Step 12: If both flagged c-pitches remaining from step 11 are members of the max-list, flag any one (and only one) former member of the min-list whose flag was removed in step 11; if both c-pitches are members of the min-list, flag any one (and only one) former member of the max-list whose flag was removed in step 11.

Step 13: Delete all non-flagged c-pitches in C

Step 14: If $N \neq 0$, N is incremented by 1 (i.e., N becomes N+1)

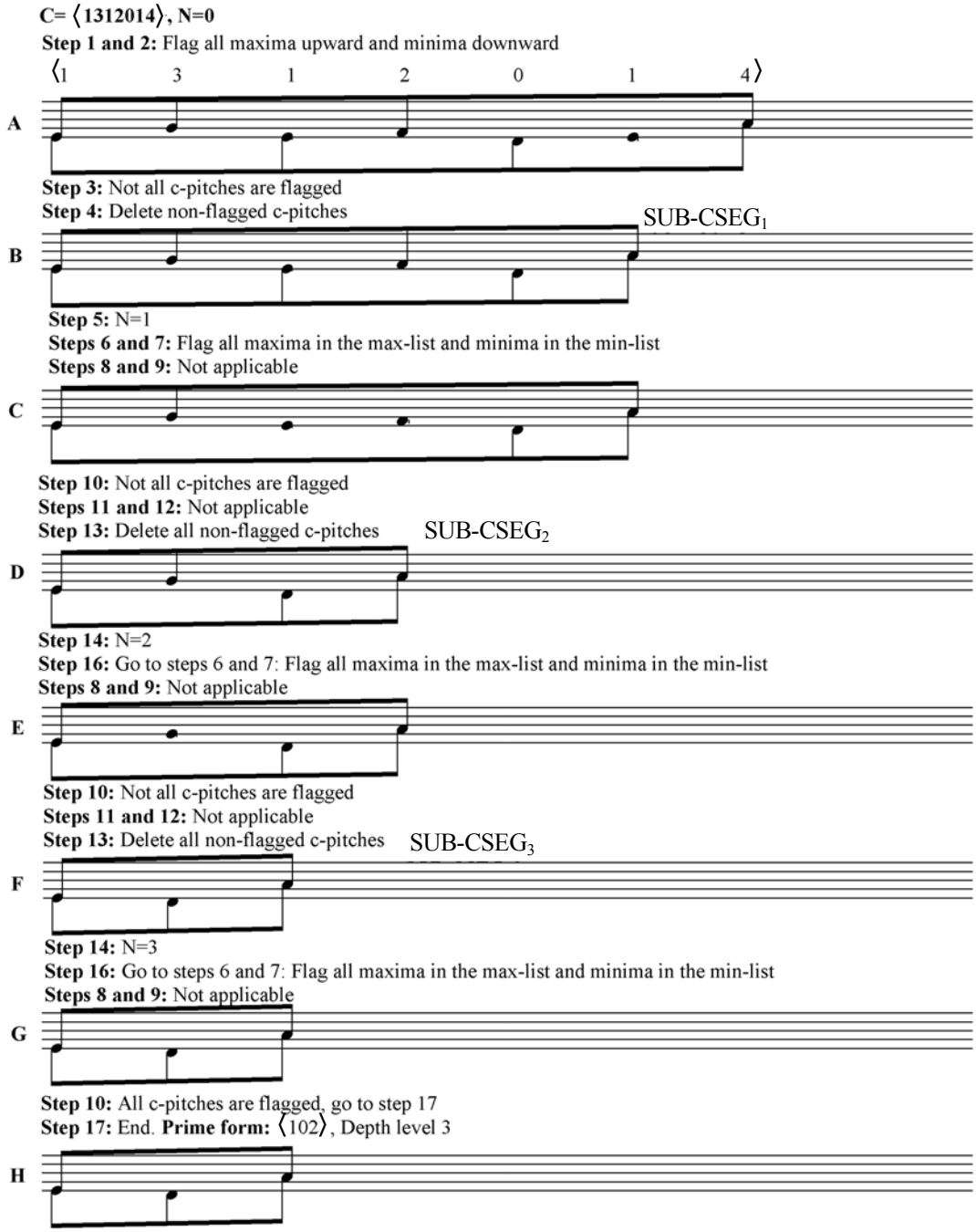
Step 15: if $N = 0$, N is incremented by 2 (i.e., N becomes N+2)

Step 16: Go to step 6

Step 17: End. N is the "depth" of the original contour C.

The reduced contour is the prime of C; if N=0, then the original C has not been reduced and is a prime itself.

Figure 2.2. Contour reduction and the sub-cseg



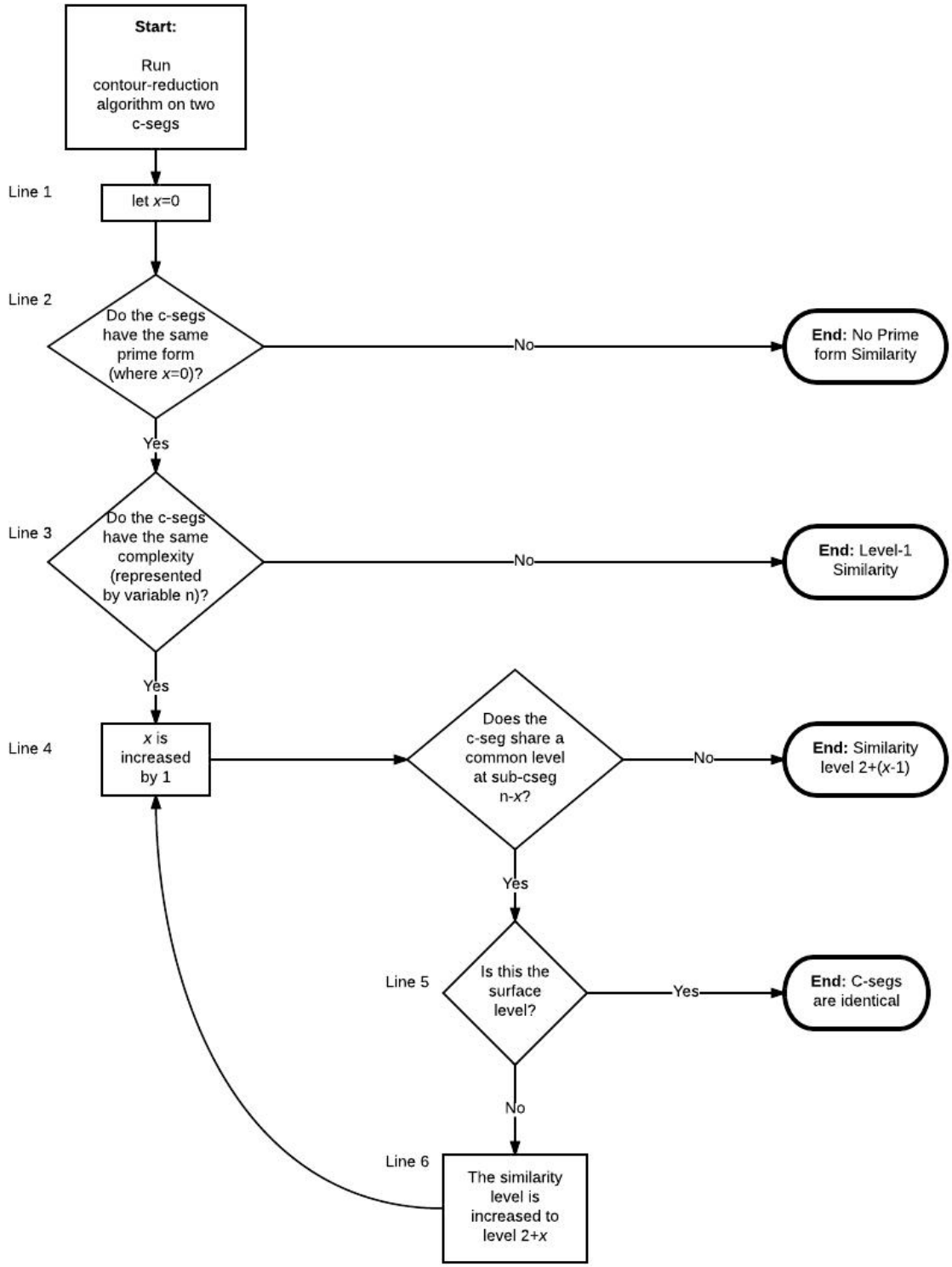
surface. Since this c-seg can be pruned further in the recursions of the algorithm, it moves on to stage two and reduces to a sub-cseg$_2$ on staff D, ⟨1203⟩. This can be pruned even further to yield a sub-cseg$_3$ of ⟨102⟩, which is the prime.

Hierarchical Levels and Similarity Relations

We can create a hierarchical procedure to determine the level of similarity using contour reduction, as I have illustrated in the flow chart shown in Figure 2.3. When two c-segs are reduced, levels are created that can be compared "level for level" with one another, (i.e. sub-cseg$_1$ with sub-cseg$_1$, etc.). This yields a threefold hierarchy of similarity. The first, most basic element of similarity to consider is the prime. If the primes of two c-segs are different, the c-segs have no levels of similarity within this system. If this is the case, the reduction can yield information about possible transformations of the c-segs (i.e. inversions, retrogrades, retrograde inversions, etc.) but little else.

If two c-segs do have the same prime, however, then these c-segs have reached level 1 similarity. From here we can move forward to a comparative concept that I will define as a c-seg's *complexity*. This variable gives us an exact count of how many unique sub-csegs exist in a given c-seg. Complexity in this sense is thought of in terms of perceptions of surface detail within a c-seg. The more intricate the surface detail is, the more iterations of the recursive algorithm we must use to arrive at the prime. That said, complexity relates directly to the notion of depth level, more specifically the depth level

Figure 2.3. The comparison-hierarchy flow chart



of prime, and therefore can be labeled with the variable N_{prime}.[8]. In essence, the greater the prime's depth level, the more complex a c-seg is. If two c-segs have reached level 1 similarity, we can then compare their complexity. If they differ in complexity, the two c-segs exist only in level 1 similarity.

If two c-segs share a level 2 similarity, we know that the two c-segs share the same prime and the same complexity. It is important to note that two c-segs must possess both of these levels in order to be compared for similarity at levels shallower than prime. To do otherwise would never yield a direct level-for-level equivalence shallower than prime. For c-segs at level 2 similarity, we can begin a comparison at the N–1 level. For c-segs that do not possess any levels shallower than prime form (or if N–1=0), level 2 similarity is the end of the comparison, and level 2 is the most similar those c-segs can be without being identical.[9]

For c-segs that do possess deeper intermediary levels between the prime and the surface, the comparison at this point becomes recursive, and we continue to evaluate the sub-csegs at these deeper levels. To execute the recursion, I have introduced the variable *x* in line 1 of the flow chart. The variable *x* represents the number of levels shallower

[8] The complexity and the depth level values are highly similar and related concepts, but the primary difference is that the complexity refers to the entire c-seg, including all of its embedded sub-csegs at various different depth levels. As such, it is labeled with the value of the depth level of prime. To use the term "depth level" in its place would be misleading, as the specific depth level of a surface-level c-seg under the MCRA would be 0. Therefore, referring to a c-seg's depth level as being anything other than 0 would be confusing. The complexity measurement clarifies this issue by removing the "depth level" label from a discussion of the global properties of the c-seg.

[9] Hypothetically, a level-3 similarity is possible, assuming that the surface level c-segs at N=0 are identical. In this instance, the c-segs would be exactly the same, and the "level-3 similarity" label would not yield any useful observations.

than prime being compared, so that N–*x* yields the specific depth level of the sub-cseg being compared at any given recursion.[10]

If two c-segs that have passed level 2 similarity possess a common sub-cseg at level n–1, they have a greater perceptual similarity. This would indicate a new level 3 similarity. The fact that the c-seg is the same on more than just the prime level would indicate that both c-segs share some of the same embellishing features of a basic prime, and we may instinctually hear these c-segs as more similar than c-segs that do not share this feature.

If the sub-csegs at N–1 are identical, the similarity level increases, and the algorithm moves on to compare sub-csegs at N–2. If these sub-csegs are identical, the similarity level increases again, and the process repeats itself at N–3, N–4 and so on, until one of two possible end conditions is met: 1) a given recursion at N–*x* does not yield a common sub-cseg, in which case the similarity level is not increased any further; or 2) one runs out of levels to compare, having gone through the iterations of the recursive comparison until reaching the surface level, in which case the c-segs are identical.

The level of similarity at this point is a representation of similarity with regard to the comparative process. For example, the label "level 3 similarity" informs us that three comparison levels are similar: the prime, the complexity, and a single level shallower than prime. The comparison ends at that point, and the two hypothetical c-segs diverge from there. Although this is important information, this label becomes more useful when placed in the context of the c-segs' complexity. A level 3 similarity would seem quite large between two c-segs that only share 4 comparable levels, yet the same level 3

[10] The variable N in this equation (as opposed to n) represents the depth level of prime.

similarity label would seem much smaller in the context of c-segs with 7 comparable levels. Therefore, it may be prudent to include a c-seg's complexity in the similarity label when applying this comparison method.

To that end, it is also useful to create a similarity index representing the degree of similarity between two c-segs as measured using this method. The similarity index compares the number of levels at which two c-segs are similar to the total number of possible levels. For example, if two c-segs share four possible levels of comparison, and they have a level 3 similarity, then the similarity index for the two c-segs would be ¾, or 0.75.[11]

Analytical Demonstration of the Hierarchical Comparison

To illustrate the fundamental components of this theory, we can apply this process to a set of c-segs from Brahms's Violin Sonata in G major, Op.78. The two c-segs occur in the violin part in measures 38–39 (Figure 2.4a) and measures 114–115 (Figure 2.4b). When comparing the c-seg of measures 38–39 with the c-seg of measures 114–115, we run the contour reduction algorithm on both c-segs, as shown in Figure 2.5c. Once the reduction has been run, we may begin our comparative analysis. The reduction is applied to both c-segs ⟨1346532140⟩ and ⟨123456789646420⟩ respectively, and pitches that are neither minima nor maxima are pruned, yielding a sub-cseg_1 of ⟨13120⟩ and ⟨14230⟩,

[11] Though the similarity index is a good measure of the total level of similarity between two c-segs, when comparing similarity indices it is important to keep in mind that different levels of complexity will lead to differences in the similarity index. An index of 1/3 and an index of 2/6 would both yield a value of 0.33, yet this does not tell us that, of the two values, one is much closer to the surface than the other. This is why it is important to keep both the similarity label and the index in mind, for the two inform each other.

respectively. Next, maxima within the max-list are pruned and minima within the min-list are pruned, producing a sub-cseg$_1$: ⟨120⟩ for both c-segs. This level produces the prime, and thus the end of the algorithm. These primes are the same, so the c-segs pass level 1 similarity. We can see that the complexity is the same: both c-segs reduce to prime at depth level 2, and thus the c-segs pass level 2 similarity. The two c-segs do not share a common sub-cseg$_1$, and therefore do not move to level 3 similarity. These c-segs remain at level 2 similarity out of 4 possible similarity levels (the fourth being the comparison of surface level c-segs), and have a similarity index of 0.5. In order to examine contour relationships at levels shallower than prime, we shall consider two more c-segs from the Brahms sonata: c-segs for measures 16 and 17 respectively, as seen in Figure 2.5a. The reduction for these c-segs is shown in Figure 2.5b. The reduction algorithm is applied to two c-segs: ⟨010342⟩ and ⟨010243⟩. After the first round of pruning, we get a sub-c-seg$_1$ of ⟨01032⟩ for both c-segs, as seen on staff B. From there, minima are pruned in the min-list, and maxima are pruned in the max-list, and we

Figure 2.4a. Score excerpt for measures 36–39

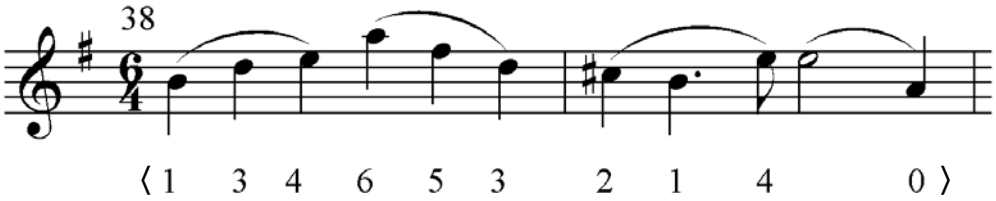


Figure 2.4b. Score excerpt for measures 114–116

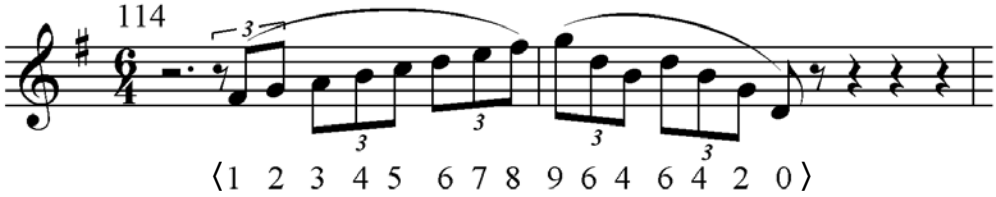
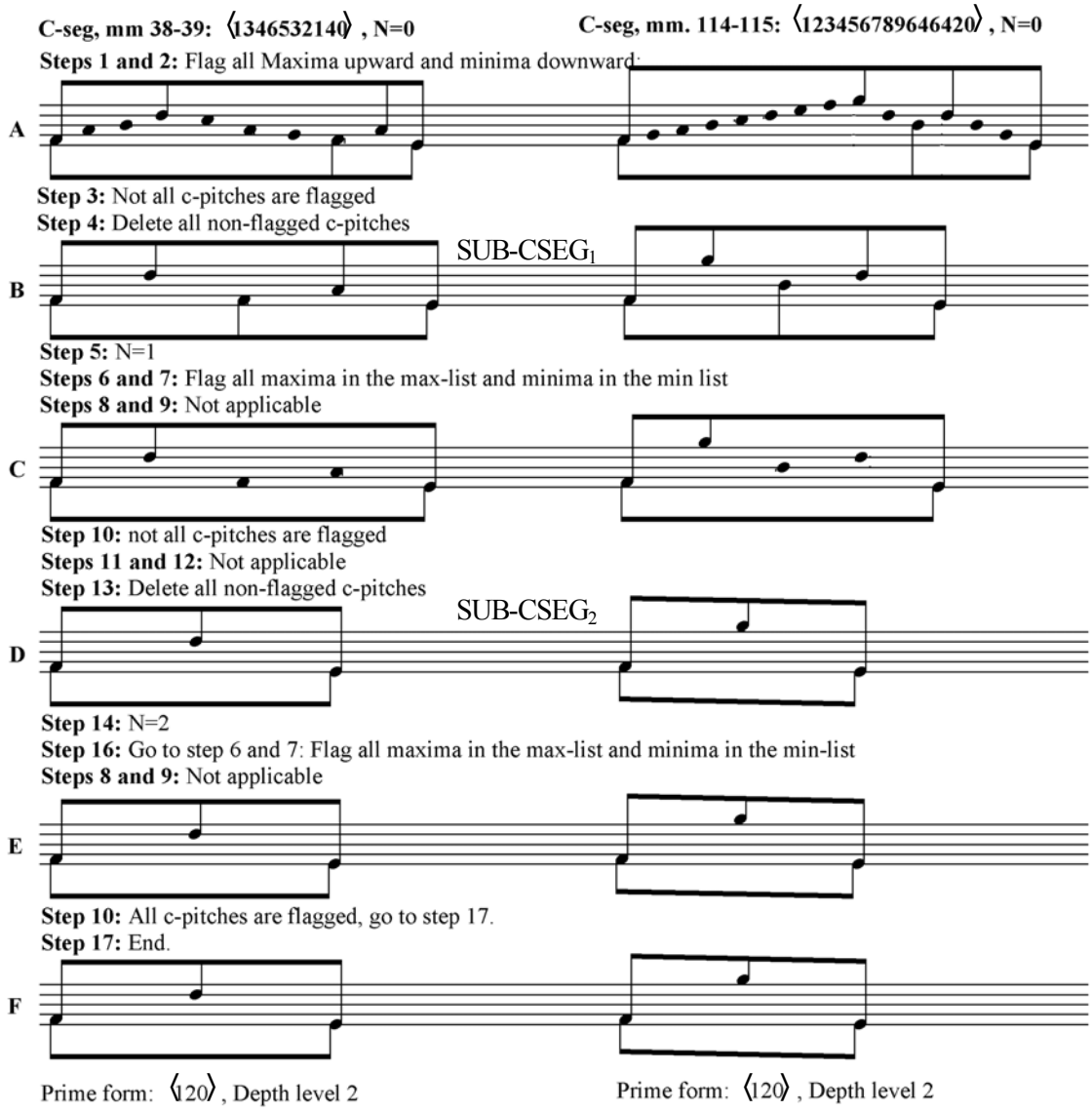


Figure 2.4c. Comparison of c-segs for measures 38–39 and measures 114–115



Note that both of these contours posses the same prime form and the same end depth level. Therefore, they share a level-2 similarity.

Figure 2.5a. Score excerpt for measures 16 and 17

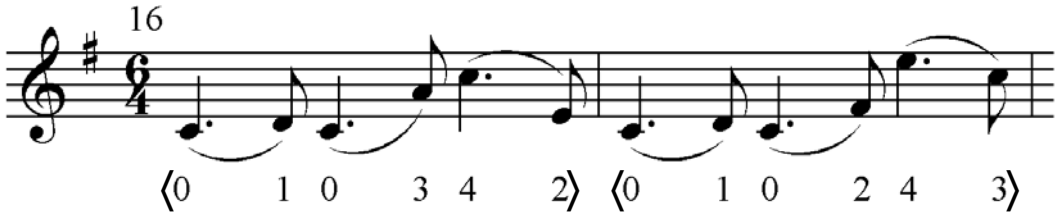
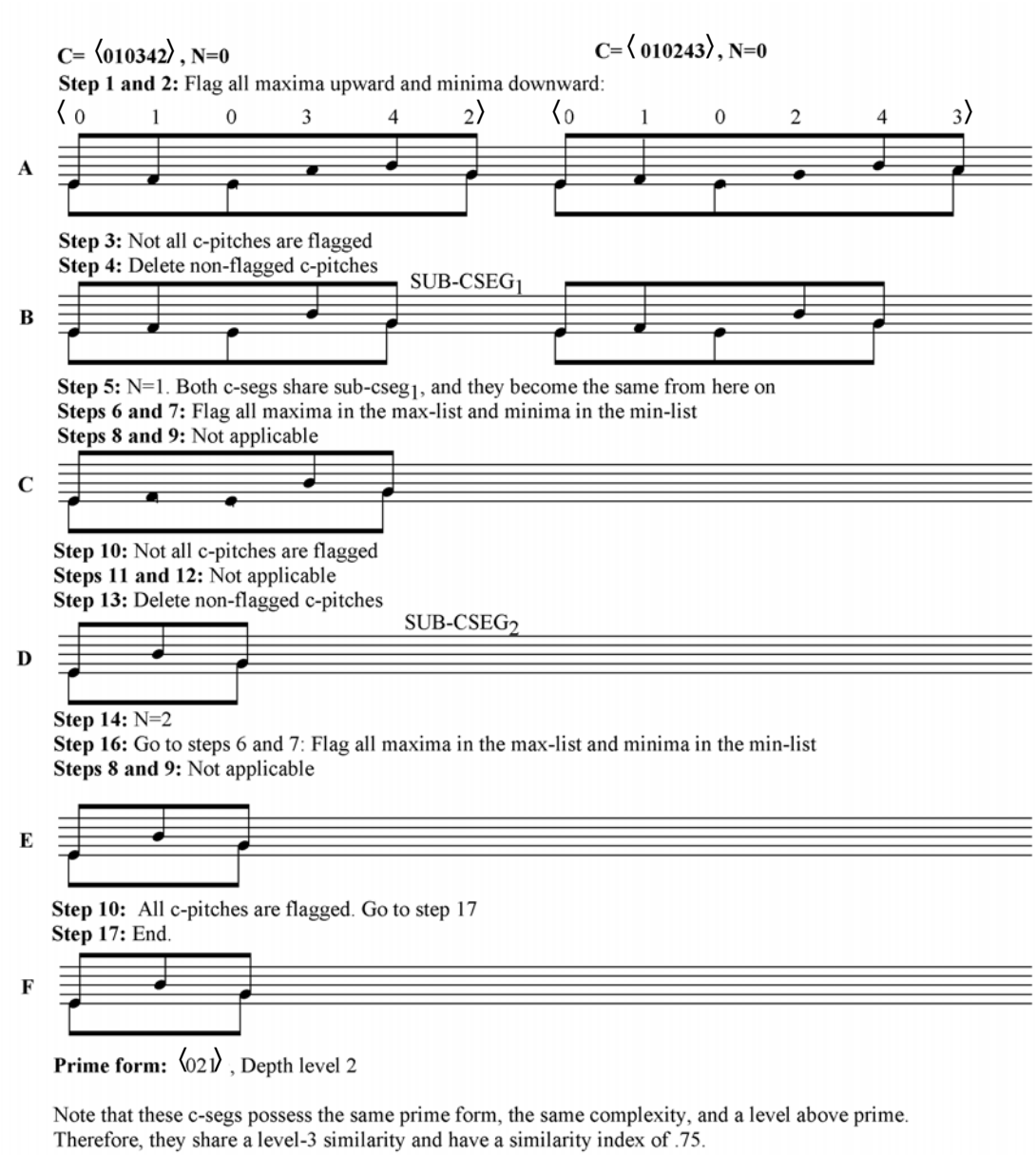


Figure 2.5b. C-seg comparison for measure 16 and measure 17

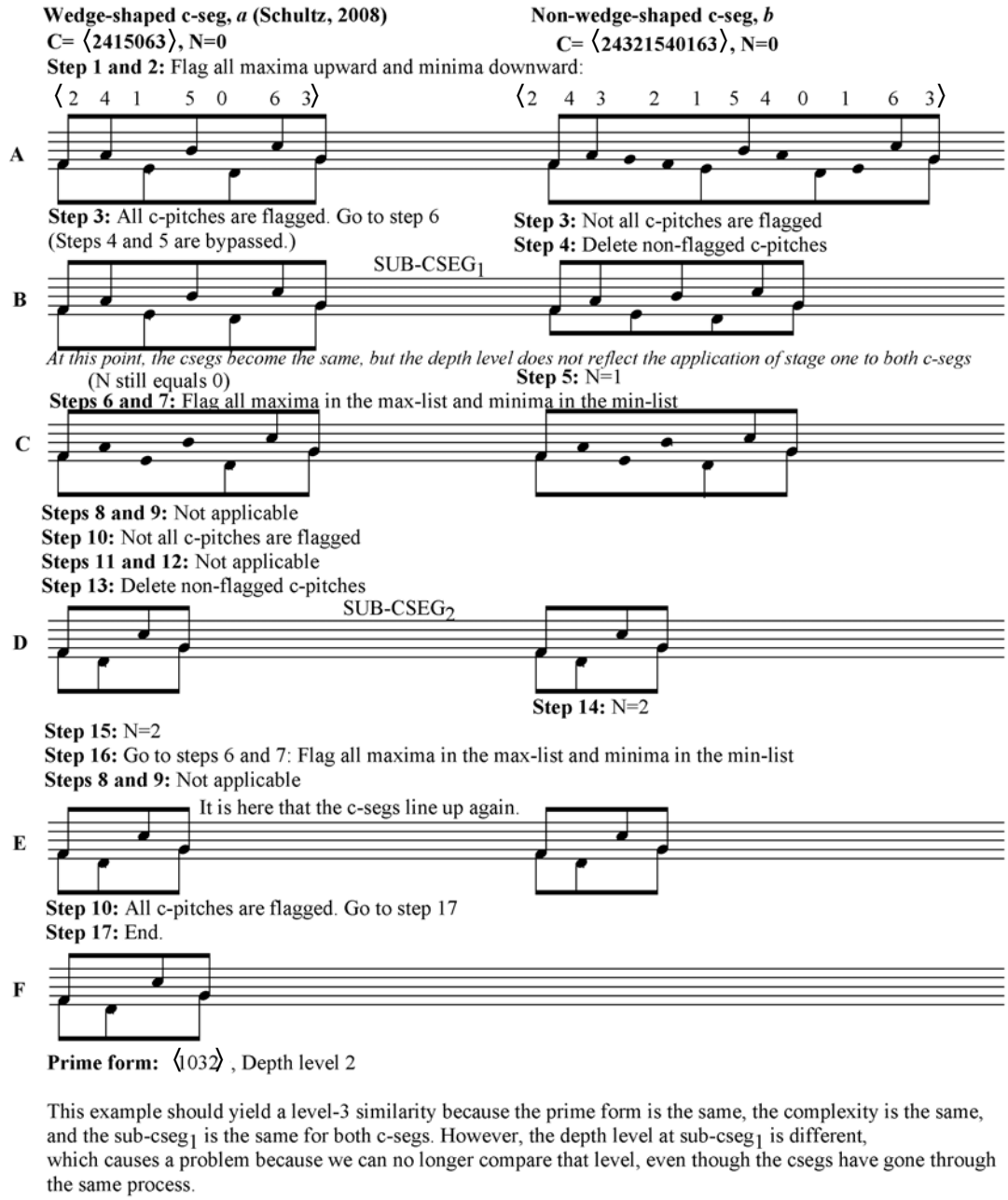


get a sub-$cseg_2$ of ⟨021⟩ for our prime in both c-segs. In this instance, we have two c-segs that meet all three levels of similarity: they possess the same prime, the same level of complexity, and they share a common level above prime. These c-segs share a level 3 similarity, out of four possible levels and have a 0.75 similarity index: the most similar these two c-segs can be without being identical. We can corroborate the high degree of similarity between these two c-segs using Marvin's CSIM function. In terms of CSIM, these c-segs at their surface levels produce a similarity quotient of 0.93, which is the highest value two six-note contours can receive without being identical.

Wedge Shapes and the N=2 Problem

The method of comparison introduced above runs into a small snag when dealing with the reduction of wedge-shaped c-segs. As Schultz (2008) has pointed out, a wedge-shaped c-seg cannot be pruned at the first stage of Morris's contour version of the algorithm because every pitch is either a maximum or a minimum. Schultz thus extended the algorithm to apply the second stage of the reduction algorithm to wedge-shaped contours. Let us consider a hypothetical example that Schultz used in his explanation of this problem, illustrated in the left column of Figure 2.6 (2008, 100). For the wedge-shaped c-seg ⟨2415063⟩, no pitches are flagged for pruning in steps 1–4, as they are all either a minimum or a maximum. Proceeding to the algorithm's second stage, however, prunes pitches in both the min- and max-lists, which produces a prime of ⟨1032⟩ and a depth level of 2, as seen on staff D.

Figure 2.6. An illustration of the wedge-shape problem



Schultz also introduced step 15, which states that if no pitches were pruned at stage 1, the depth level would not increase to depth level 1, but skip straight to depth level 2 as the rest of the algorithm is applied. However, when attempting to compare c-segs at depth levels shallower than prime, the depth levels do not line up conceptually

with each other. In a non-wedge-shaped contour, stage two reductions are always applied to a sub-cseg$_1$, generated after depth level 1. However, a wedge-shaped contour in the current state of the algorithm would reach stage two reduction at a depth level of 0. We should be able to compare these c-segs along with the other c-segs, but we cannot conceptually do that until the depth levels are properly aligned.

Figure 2.6 illustrates the problem. The first stage of the algorithm yields no pruned pitches for the c-seg *a* in the left column, while pitches are pruned for c-seg *b* in the right column. When the two contours proceed to step 6 of the algorithm, c-seg *a* flows from step 3 to step 6, and c-seg *b* flows through steps 4 and 5. At this point both c-segs arrive at the second stage of the algorithm, but with different depth levels. This seems to suggest that the first stage was not applied to c-seg *a* even though it did go through that initial pruning process. Furthermore, the pruned c-seg *b* arrives at the second stage identical to c-seg *a*. This suggests a similarity on some level, but it is unclear as to precisely which one. These c-segs should yield a level 3 similarity, because their primes, complexity, and sub-cseg$_1$s are all the same. However, their depth level N values at sub-cseg$_1$ differ, which is problematic because we can no longer compare that level, even though the c-segs have gone through the same process. As it stands now, the two c-segs become the same only after step 16 on staff E, which forbids the possibility of a deeper level of similarity. In other words, the current algorithm would yield only a level 2 similarity between these c-segs, while a closer examination of the c-segs would suggest that a level 3 similarity would be more appropriate.

I see two possible ways to fix this problem. The first solution (illustrated in Figure 2.7a) is to simply remove steps 3 and 15 (the steps governing the repetition of

N=0), and state that depth levels change regardless of whether or not pitches have been pruned in the first stage. Figure 2.7b shows the reduction from Figure 2.6, corrected according to the steps in Figure 2.7a to allow the levels shallower than prime to line up accurately. The other solution (shown in Figure 2.8a) is not to allow depth levels to be stage specific. For instance, if no pitches were pruned in stage 1, we are still left with a depth level 0 c-seg, and we move on to stage 2, where the N=0 c-seg becomes an N=1 c-seg instead of N=2. Figure 2.8b shows the reduction from Figure 2.6 using the second possible correction, and illustrates the conceptual problems with that solution: in this example, the depth levels have been modified to reflect that no action was taken in stage 1. These two c-segs now share only a prime, since the algorithm has been modified in such a way that the depth levels do not reflect the difference between a level at stage 1 and a level at stage 2.

There are small difficulties with each of these solutions, just as there are with the solution described by Schultz. The first solution suggests that depth level 1 is somehow different from depth level 0, whereas the second solution struggles to reflect the differences between stage 1 reduction and stage 2 reduction and thus has difficulty reflecting the deeper background c-seg. For the sake of comparing each level at each stage and step of the reduction process, the first solution I have suggested may be the most acceptable. It would indicate that both c-segs have gone through both stages of the reduction, and still retain the deeper-level representation necessary to differentiate stage 1 from stage 2. In other words, one cannot skip the first stage just because no pitches would be pruned. One still must apply that stage, and the depth level numbering should reflect this.

Figure 2.7a. The Modified Contour Reduction Algorithm[12]
Algorithm: Given a contour C and a variable N:

Step 0: Set N to 0

Step 1: Flag all maxima in C upwards; call the resulting set the *max-list*

Step 2: Flag all minima in C downwards; call the resulting set the *min-list*

OMITTED: ~~(Step 3: If all c-pitches are flagged, go to step six)~~

Step 4: Delete all non-flagged c-pitches in C

Step 5: N is incremented by 1 (i.e., N becomes N+1)

Step 6: Flag all maxima in the max-list upwards. For any string of equal and adjacent maxima in the max-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 7: Flag all minima in the min-list downwards. For any string of equal and adjacent minima in the min-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 8: For any string of equal and adjacent maxima in the max-list in which no minima intervene, remove the flag from all but (any) one c-pitch in the string.

Step 9: For any string of equal and adjacent minima in the min-list in which no maxima intervene, remove the flag from all but (any) one c-pitch in the string.

Step 10: If all c-pitches are flagged, and no more than one c-pitch repetition in the max-list and min-list (combined) exists, not including the first and last c-pitches of C, proceed directly to step 17.

Step 11: If more than one c-pitch repetition in the max-list and/or min-list (combined) exists, not including the first and last c-pitches of C, remove the flags on all repeated c-pitches except those closest to the first and last c-pitches of C.

Step 12: If both flagged c-pitches remaining from step 11 are members of the max-list, flag any one (and only one) former member of the min-list whose flag was removed in step 11; if both c-pitches are members of the min-list, flag any one (and only one) former member of the max-list whose flag was removed in step 11.

Step 13: Delete all non-flagged c-pitches in C

Step 14: If $N \neq 0$, N is incremented by 1 (i.e., N becomes N+1)

OMITTED: ~~(Step 15: if N = 0, N is incremented by 2 (i.e., N becomes N+2))~~

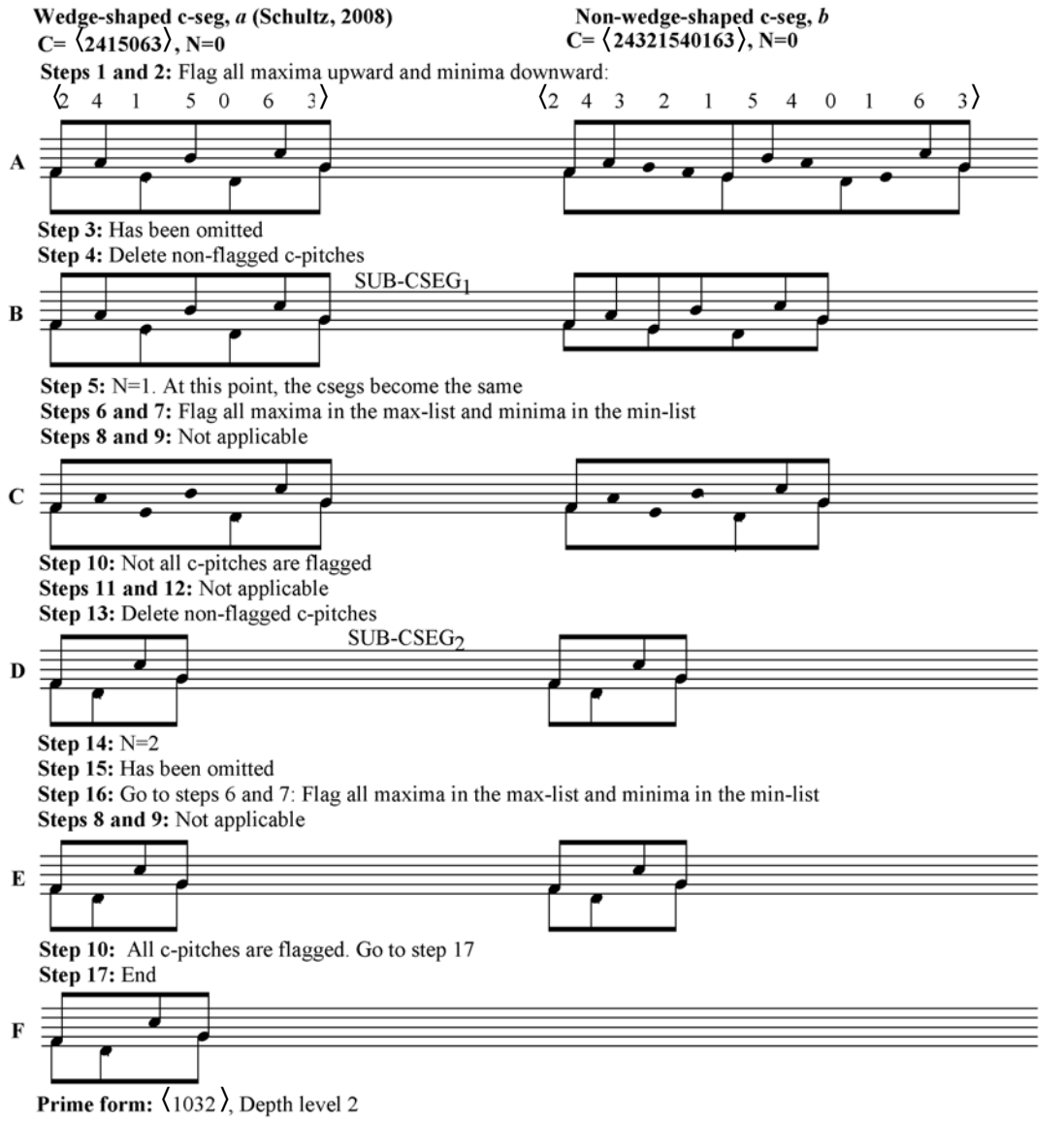
Step 16: Go to step 6

Step 17: End. N is the "depth" of the original contour C.

The reduced contour is the prime of C; if N=0, then the original C has not been reduced and is a prime itself.

[12] Original algorithm shown in Schultz 2008, p. 108.

Figure 2.7b. A correction of the wedge-shape problem



In this example, I have corrected the problem seen in Example 7. The depth levels now match up with one another, and we can now conduct a proper comparison at those levels above prime. This example now illustrates a level-3 similarity, wherein the two c-segs share prime form, complexity, and sub-cseg$_1$.

Figure 2.8a. The Alternate Modification of the Algorithm[13]
Algorithm: Given a contour C and a variable N:

Step 0: Set N to 0

Step 1: Flag all maxima in C upwards; call the resulting set the *max-list*

Step 2: Flag all minima in C downwards; call the resulting set the *min-list*

Step 3: If all c-pitches are flagged, skip to step 5.

Step 4: Delete all non-flagged c-pitches in C

Step 5: if no c-pitches were pruned in step 4, N=0. If any c-pitches were pruned in step 4, N is incremented by 1 (i.e., N becomes N+1)

Step 6: Flag all maxima in the max-list upwards. For any string of equal and adjacent maxima in the max-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 7: Flag all minima in the min-list downwards. For any string of equal and adjacent minima in the min-list, flag all of them, unless: (1) one c-pitch in the string is the first or last c-pitch of C, then flag only it; or (2) both the first and last c-pitches of C are in the string, then flag (only) both the first and last c-pitches of C.

Step 8: For any string of equal and adjacent maxima in the max-list in which no minima intervene, remove the flag from all but (any) one c-pitch in the string.

Step 9: For any string of equal and adjacent minima in the min-list in which no maxima intervene,

remove the flag from all but (any) one c-pitch in the string.

Step 10: If all c-pitches are flagged, and no more than one c-pitch repetition in the max-list and min-list (combined) exists, not including the first and last c-pitches of C, proceed directly to step 17.

Step 11: If more than one c-pitch repetition in the max-list and/or min-list (combined) exists, not including the first and last c-pitches of C, remove the flags on all repeated c-pitches except those closest to the first and last c-pitches of C.

Step 12: If both flagged c-pitches remaining from step 11 are members of the max-list, flag any one (and only one) former member of the min-list whose flag was removed in step 11; if both c-pitches are members of the min-list, flag any one (and only one) former member of the max-list whose flag was removed in step 11.

Step 13: Delete all non-flagged c-pitches in C

Step 14:N is incremented by 1 (i.e., N becomes N+1)

OMITTED~~: (Step 15: if N = 0, N is incremented by 2 (i.e., N becomes N+2))~~

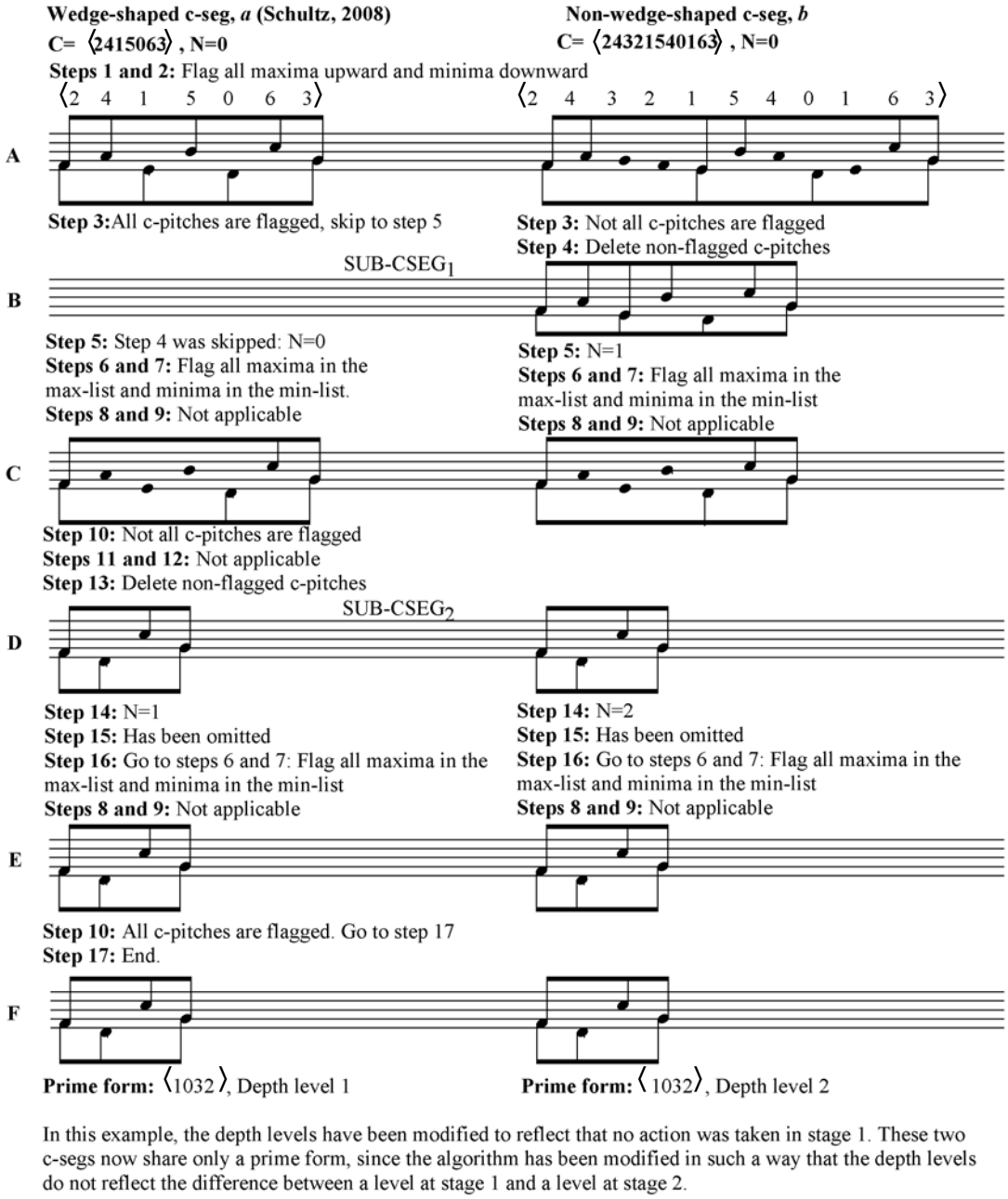
Step 16: Go to step 6

Step 17: End. N is the "depth" of the original contour C.

The reduced contour is the prime of C; if N=0, then the original C has not been reduced and is a prime itself.

[13] Original algorithm shown in Schultz 2008, p. 98.

Figure 2.8b. The alternate modification of the wedge-shape problem



In this example, the depth levels have been modified to reflect that no action was taken in stage 1. These two c-segs now share only a prime form, since the algorithm has been modified in such a way that the depth levels do not reflect the difference between a level at stage 1 and a level at stage 2.

Further Analytical Applications of This Comparative Theory

Having presented the concept of contour similarity using the reduction algorithm, I will now briefly discuss its possible analytical applications. The types of comparisons described above can reveal differences in the treatment of motives in tonal music from a

contour-based perspective. We may use the theory to see just how similar two c-segs are on a deeper level, one that is perhaps less immediately salient. In addition, this method of comparison can give us additional information needed to differentiate c-segs that have been compared in other ways, such as with Marvin's CSIM function. We can also address the idea of contour as a motive and look at how composers include those contours in different ways throughout a piece.

One can find many comparisons among different segmentation levels that suggest a matryoshka-like nesting of contours within one another. For instance, Figure 2.9a displays the first six measures of the Brahms Sonata. At the level of the motive, the primes are ⟨10⟩, ⟨01⟩, ⟨201⟩, and ⟨021⟩. However, construing the ⟨10⟩ and ⟨01⟩ segments together as a single unit yields a prime of ⟨201⟩, the same prime as the motive immediately following in m. 5, as well as the same prime as the second two motives (⟨201⟩ and ⟨012⟩) combined. Figure 9b shows that while these two contours have the same prime, their surface-level contours are in fact quite different. No depth levels above prime match up, and even the depth levels of the primes themselves are different. These two contours share only a level 1 similarity, suggesting a low level of similarity as represented by a similarity index of only 0.2. In short, Brahms's treatment of this prime form is vastly different in these two instances.

Figure 2.9a. Score example of measures 1–9

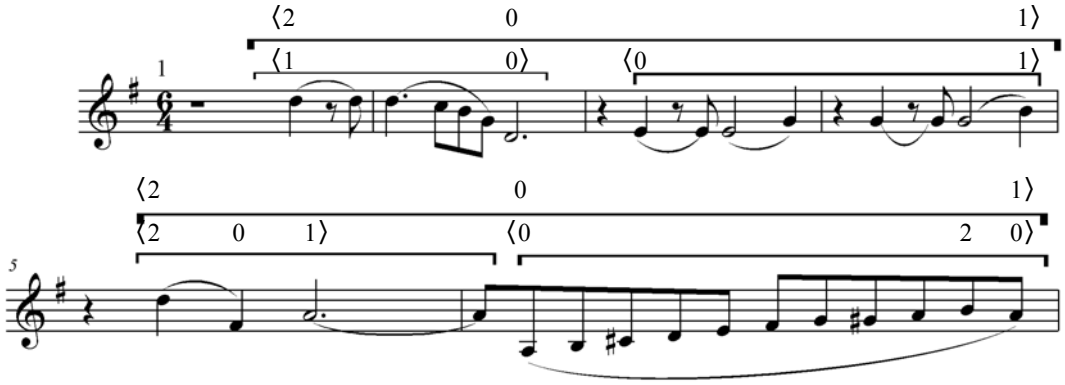
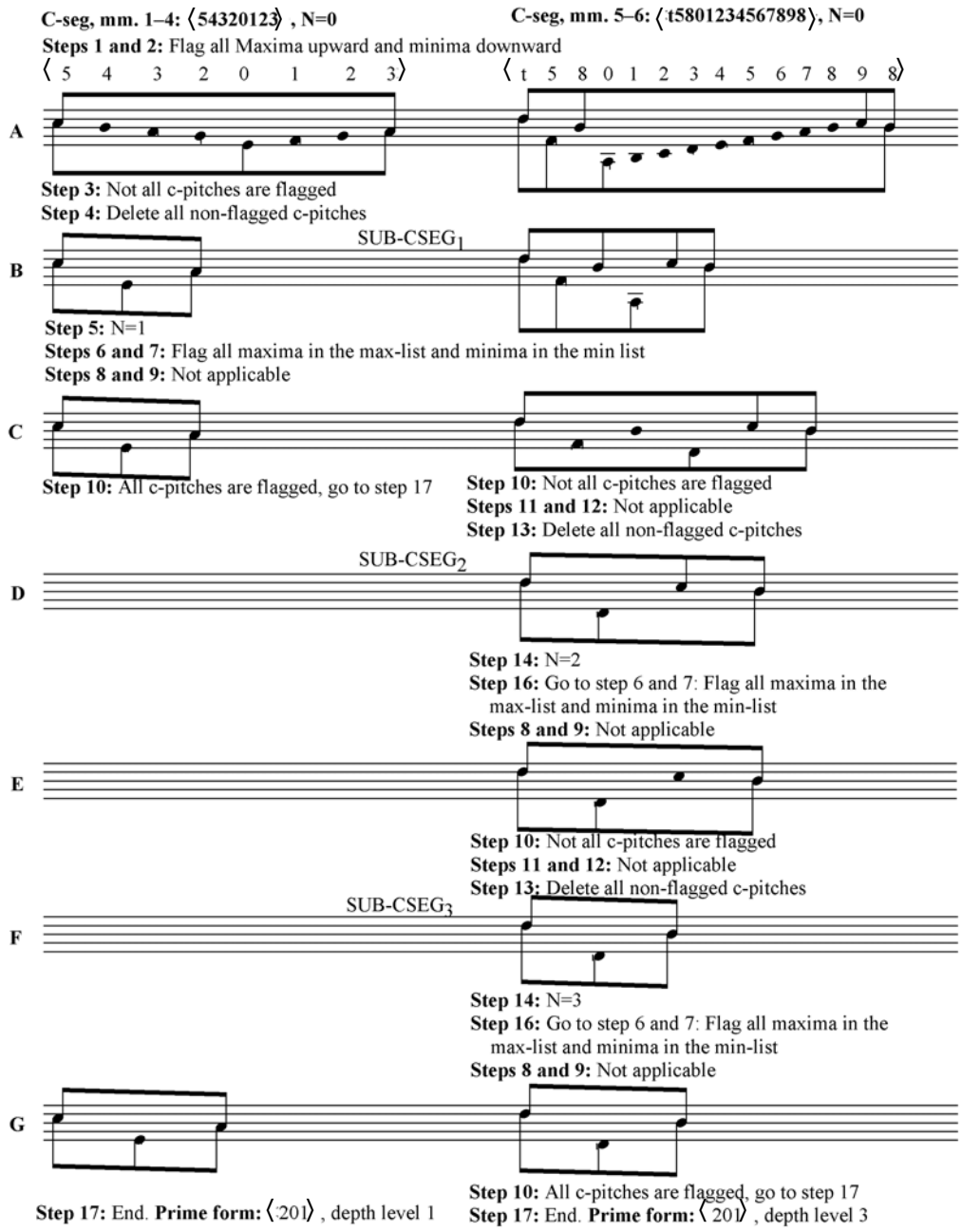
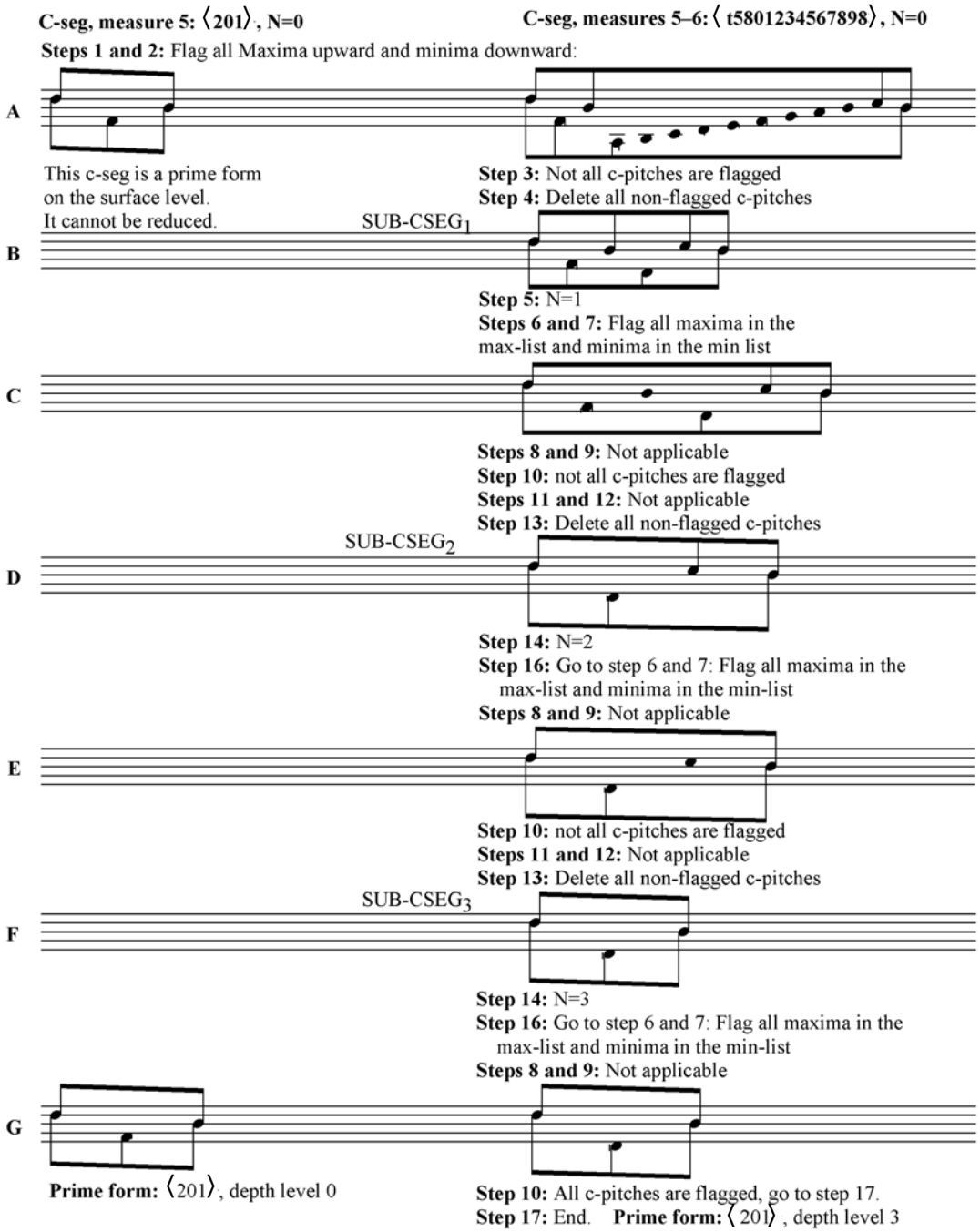


Figure 2.9b. Comparison of c-segs for measures 1–4 and measures 5–6



An example of a nested c-seg can be seen in measures 5 and 6, the score for which can be found in Figure 2.9a. Figure 2.10 shows that the c-seg for measure 5,

Figure 2.10. Comparison of c-segs for measure 5 and measures 5–6



⟨201⟩, is a prime in itself at depth level 0. The c-seg for measures 5 and 6 together is also ⟨201⟩ at a deeper depth level. This combined c-seg is more complex, as indicated by its depth level. The comparison shows us Brahms's different treatment of the prime sub-cseg, and can show us how smaller c-segs at shallower depth levels can be combined to create a more complex contour with the same prime.

Conclusion

This chapter has shown that c-segs can be compared on multiple levels using the contour reduction algorithm. The reduction algorithm creates a hierarchy of comparison levels that illustrate an increase in similarity with a corresponding increase in level. A level 1 similarity corresponds to c-segs that share only the same prime. A level 2 similarity indicates that two c-segs share the same complexity, as represented by depth level equivalence. Level 3 similarity indicates that two c-segs share a common sub-cseg, and level 4-and-above similarity indicates that two c-segs share multiple common sub-csegs. These levels of comparison can give us insight into the comparison of c-segs on different levels, and can also give us a new tool that we can use to further compare c-segs that have already been compared using other methods. In the case of the Brahms c-segs, this comparative method was able to show varying degrees of similarity among c-segs that all possessed the same prime. Since some c-segs are more similar than others, as represented by the intermediary sub-csegs, pointing out the similarity of these intermediary sub-csegs becomes a useful strategy for refining the comparative analysis using the MCRA. In general, contour reduction is a useful set of tools to have in one's analytical toolbox because it can describe the similarity of c-segs at different levels.

CHAPTER 3

EXTENSIONS TO THE COMPARATIVE PROCESS

The method of comparison I have introduced in the previous chapter illustrates the usefulness of the MCRA in determining finer gradations of structural similarity. One especially advantageous aspect of this approach is its ability to reduce and compare c-segs of different cardinalities. However, the approach is specific only to c-segs that reduce to the same prime at the same depth level, and has no way of comparing c-segs with different complexities, as measured by the differing depth levels of prime. In this chapter, I will focus on this issue, and present a more generalized version of the comparison methodology of the preceding chapter that can account for this phenomenon.

Primes on Differing Depth Levels

Melodic contour plays an important role in the structure of plainchant, yet a rigorous analysis of contour in plainchant has not yet occurred. Because of the ability of the MCRA to systematically reduce a c-seg of any cardinality to its prime, and the ability of the comparative process to use the hierarchy produced by the MCRA to describe similarity, these tools are ideal for use on this particular repertoire. The MCRA is particularly good for finding the deeper structure of longer c-segs, including long melismatic passages, and phrase-length melodies. Such c-segs can be subdivided into smaller c-segs to show motivic relationships between the smaller and larger structures. This type of relationship is apparent especially in certain types of plainchant, namely the Alleluia. Consider the verse of *Alleluia Cantate domino*, shown in Figure 3.1a and Figure 3.1b. The prime of the verse is ⟨021⟩ at a depth level of N=5. Within this verse,

smaller instances of the ⟨021⟩ prime occur. Figure 3.1c shows the first half of the phrase, "cantate domino," which also has a ⟨021⟩ prime at a shallower level: N=4. At an even smaller scale of segmentation, the c-seg set to the word "cantate" also has a prime of ⟨021⟩ and a depth of 4. Such relationships occur at varying degrees of similarity when each of these is compared using the hierarchical comparison method.

However, a unique condition begins to arise when one analyzes c-segs on a larger scale in this fashion. As longer c-segs are reduced, the chances of having extremely complex contours rise. When this happens, there is a greater probability of having identical sub-csegs at differing depth levels. Consider Figure 3.2, which shows two phrases from related chants: *Alleluia Dies sanctificatus* (c-seg *a*) and *Alleluia Hic est discipulus* (c-seg *b*). As displayed in the figure, c-seg *a* has a greater degree of complexity, reducing to prime at N=5, while c-seg *b* reduces to prime only at N=4. Despite this difference, both c-segs feature a common sub-cseg: ⟨1020⟩, but at differing depth levels. Such depth level displacement presents problems for the direct comparison of sub-csegs using the hierarchical comparison.

Figure 3.1a. Alleluia Cantate Domino

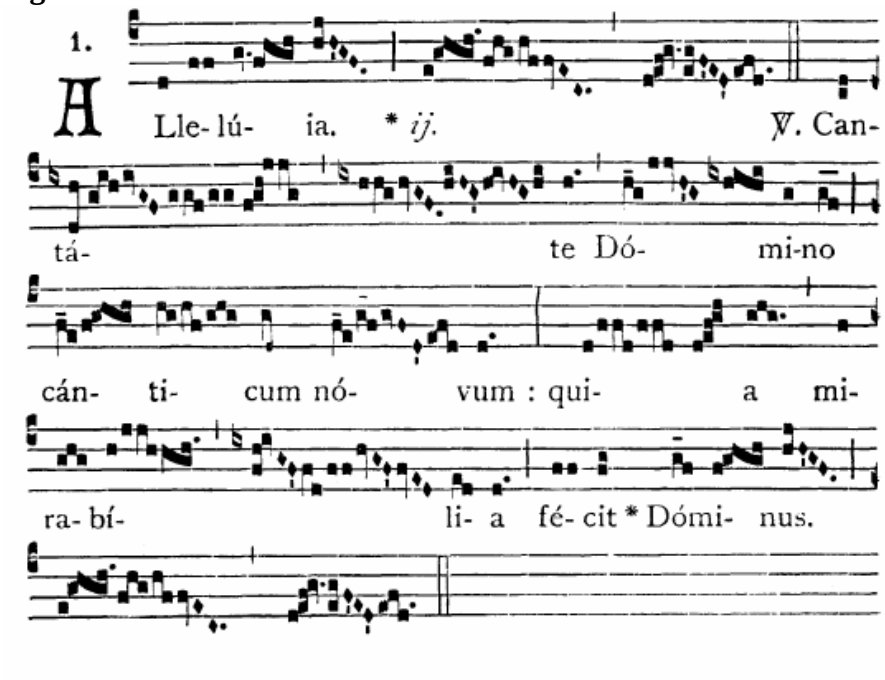


Figure 3.1b. Reduction of the verse from *Alleluia Cantate Domino*

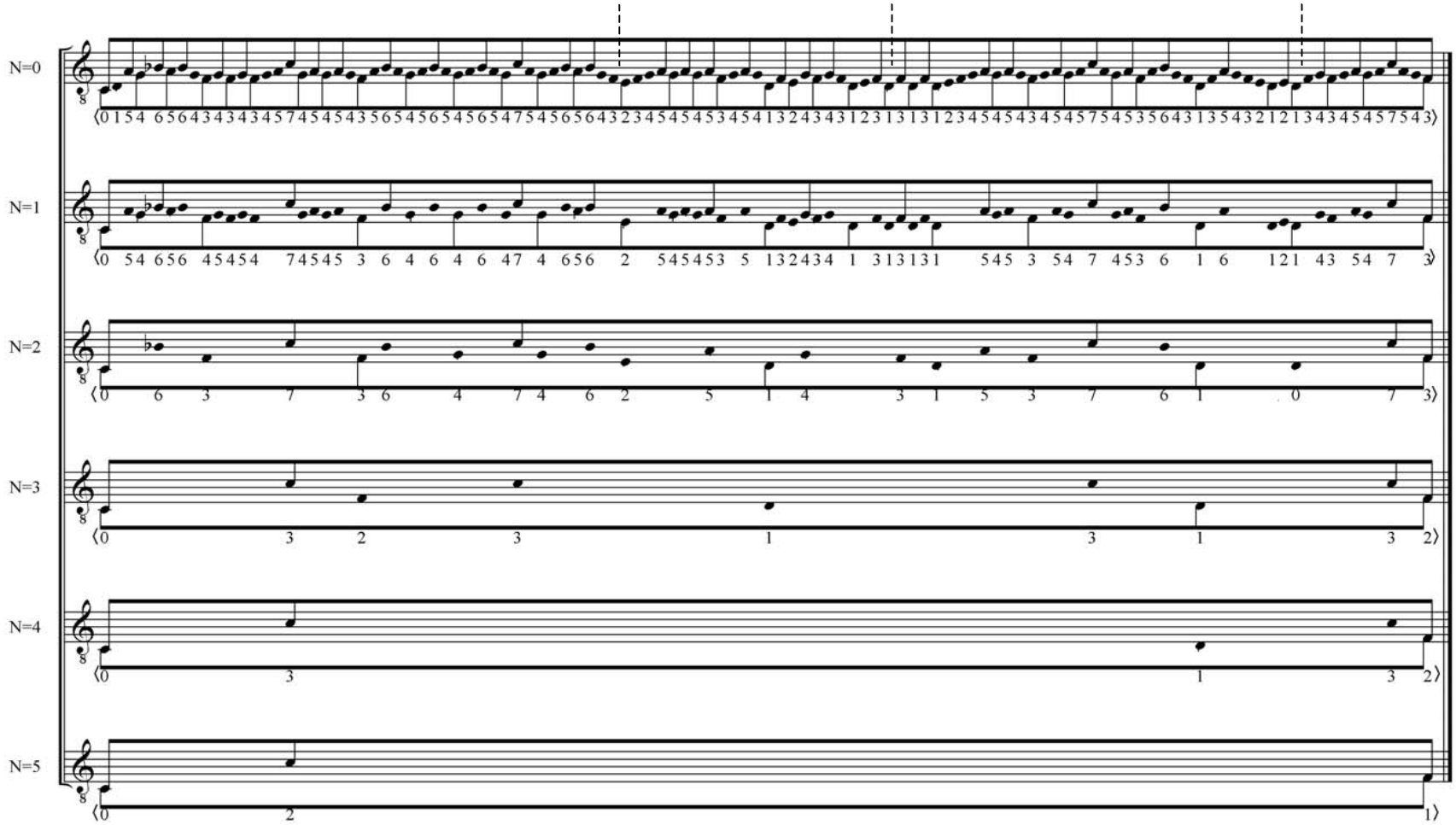


Figure 3.1c. Reduction of the phrase “cantate domino”

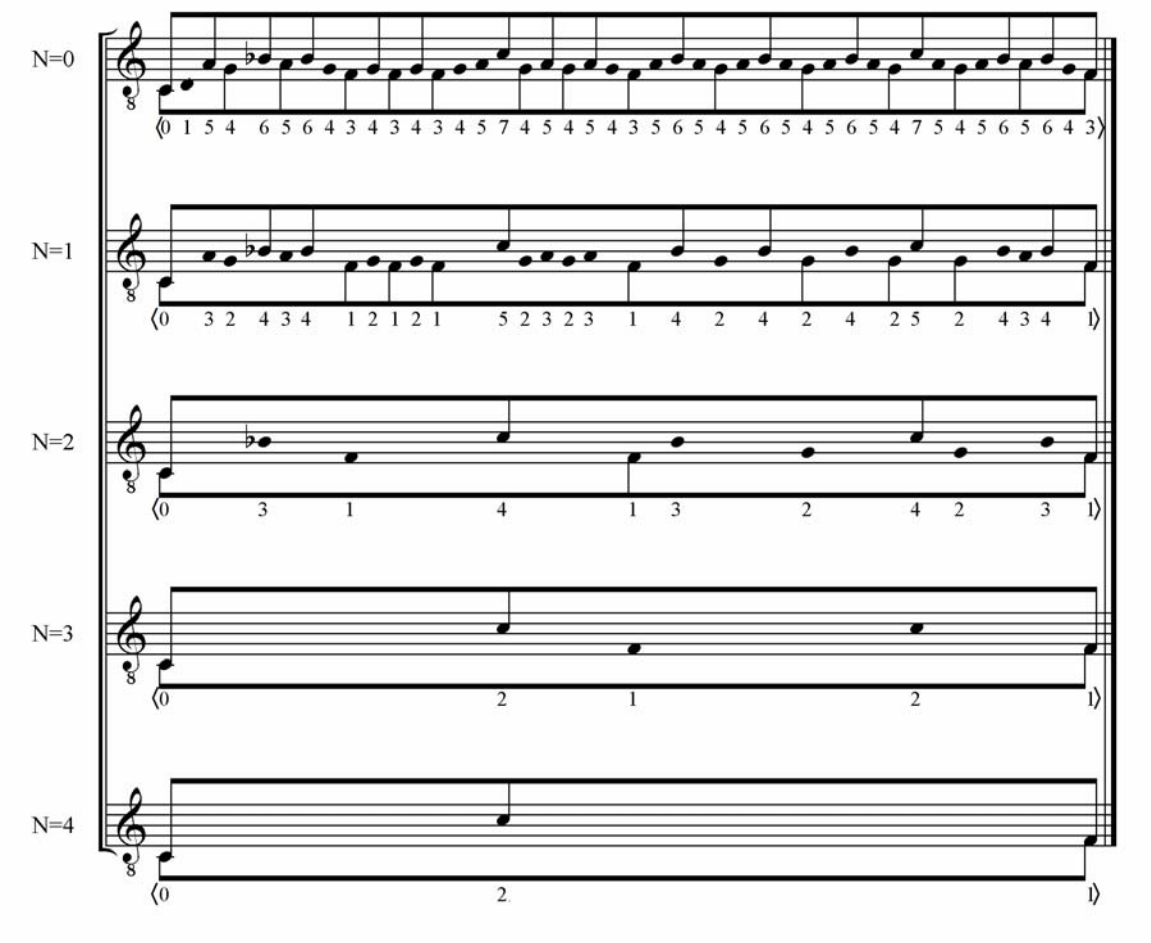


Figure 3.1d. Reduction of the c-seg on the word "cantate"

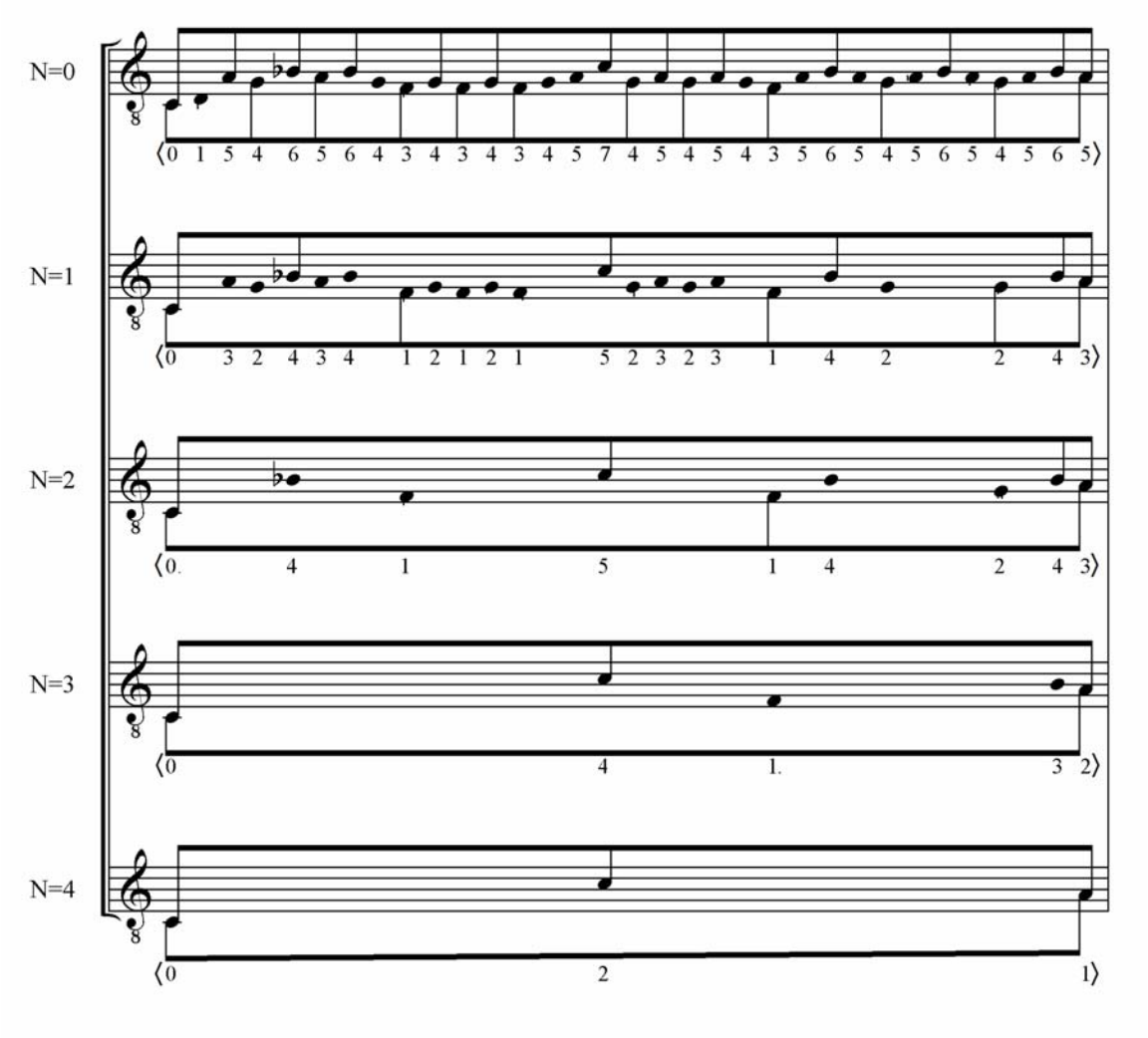


Figure 3.2a. Phrases from Alleluia Dies Sanctificatus and Alleluia Hic Est Discipulus

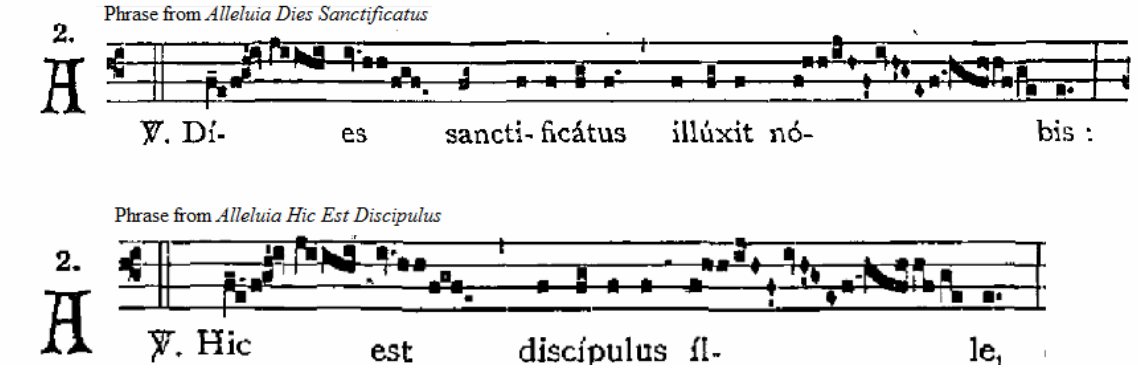
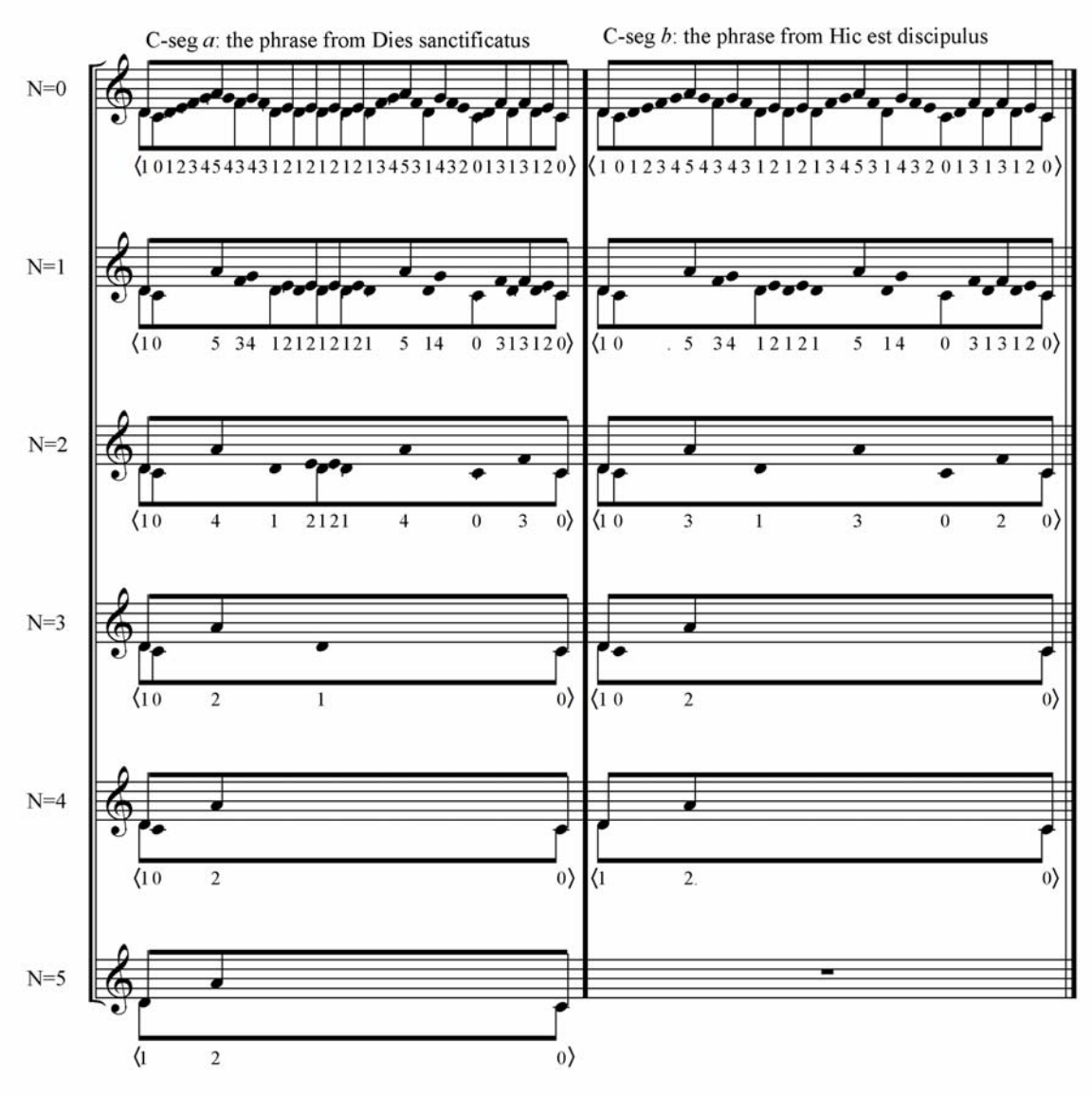


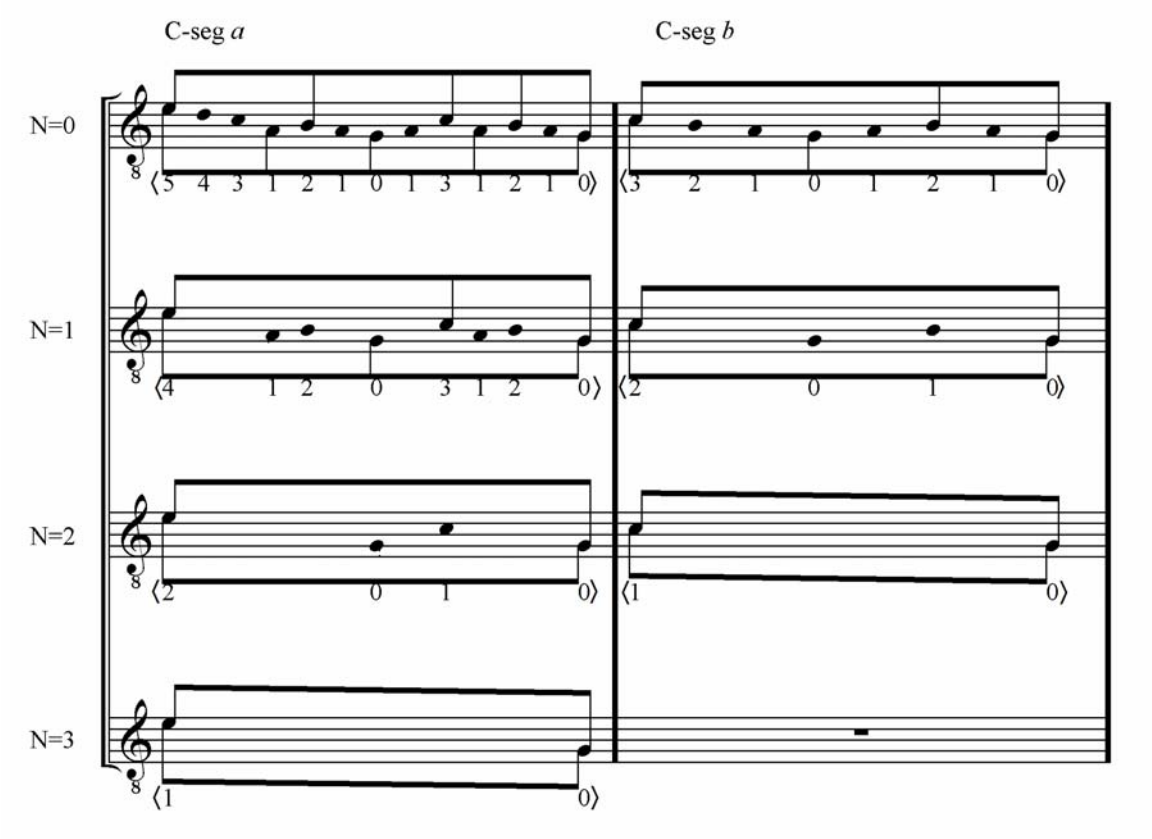
Figure 3.2b. Reductions of the phrases presented in Figure 3.2a.



The Displacement Problem

Figure 3.3 shows another instance of this issue. Two hypothetical c-segs are shown, which both feature a ⟨10⟩ prime, at respective depth levels of N=3 and N=2. According to the current comparative method, this would suggest a level 1 similarity because the primes are identical, but occur at different depth levels.

Figure 3.3. Reduction of hypothetical c-segs to show displacement



The difficulty arises when one examines the sub-csegs within each of the c-segs, produced by the algorithm. Both c-segs include a sub-cseg of ⟨2010⟩, yet they occur at different depth-levels, and therefore are not compared using the method outlined in the previous chapter. The comparison process supposes that sub-csegs are comparable only in pairs of c-segs with the same complexity. However, the c-segs presented in Figure 3.3 do not fit this definition. In looking at the two lists of sub-csegs, it becomes clear that the relationship between these two c-segs may be stronger than previously stated, due to the common occurrence of the ⟨2010⟩ sub-cseg. Our current notion of the comparison process would end the comparison at level 1 similarity due to the location of the complexity measurement. Under the process, no intermediary levels are compared, yet

the possibility of similarity on these levels suggests that the comparative process must be modified in order to account for differences of complexity in c-segs such as these.

Because the notion of complexity is so important as a determinant of similarity within the direct comparison process, an alternate calculation must be introduced in order to account for comparisons between sub-csegs at different depth levels. I propose a new concept, the displaced similarity, which measures both the number of similar sub-csegs and the depth displacement between corresponding sub-csegs.

The Displacement Comparison

C-segs of differing complexity require different procedures to outline the level of similarity. Because the complexity differs, the similarity between two c-segs is more obscured than c-segs of the same complexity. One can explain this principle using the concept of *displacement*: the degree with which a prime level similarity is obscured by differences in complexity. The level of displacement will therefore reflect the mathematical difference between the levels of complexity. One can think of this type of displacement as similar to the generational displacement between cousins at various levels of removal. For example, in a relationship between first cousins, once removed, one cousin is separated from the common ancestor by a generation. In this sense, if one c-seg has an extra sub-cseg that is not present in the second c-seg, then they have a displacement of 1: they are like the cousins once removed.

Turning back to Figure 3.3, the prime of c-seg *b* is the same as the prime of c-seg *a*, exactly one depth-level shallower. Since the primes are identical, and the depth level is different, we can state that the prime of the shallower c-seg has been displaced by one

level. Figure 3.4 again shows another instance of c-segs with the same prime form. Here, c-seg *a* has a complexity of 3, c-seg *b* has a complexity of 2, and c-seg *c* has a complexity of 1. The primes of c-seg *a* and c-seg *b* are only one depth-level apart, and therefore have a displacement of 1. However, the primes of c-seg *a* are two depth-levels apart, so they have a displacement of 2. We can use this displacement measurement to create a series of steps that will allow the comparison of sub-csegs needed in Figure 3.2. The direct comparison of depth-levels with no displacement represents the closest possible similarity. The displacement therefore will reflect the notion that the c-segs are now less similar.

Figure 3.4. Reductions of c-segs with ⟨021⟩ occurring at different depth levels

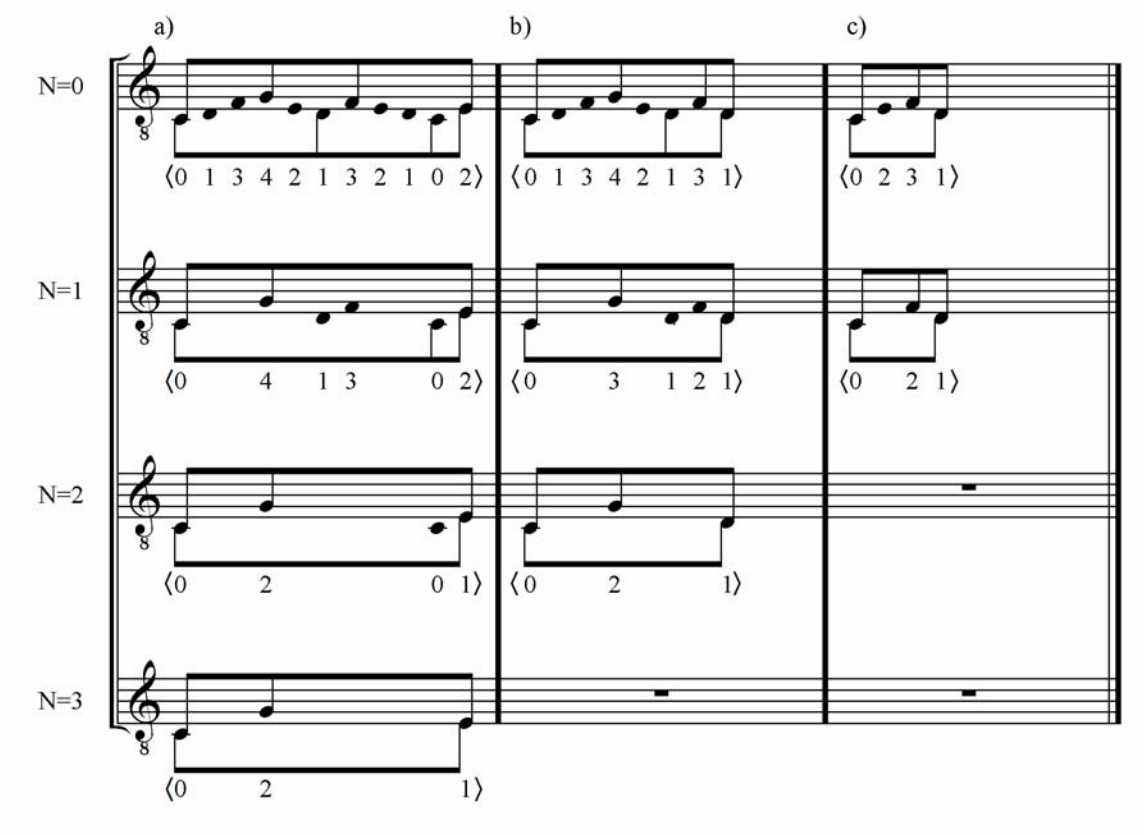
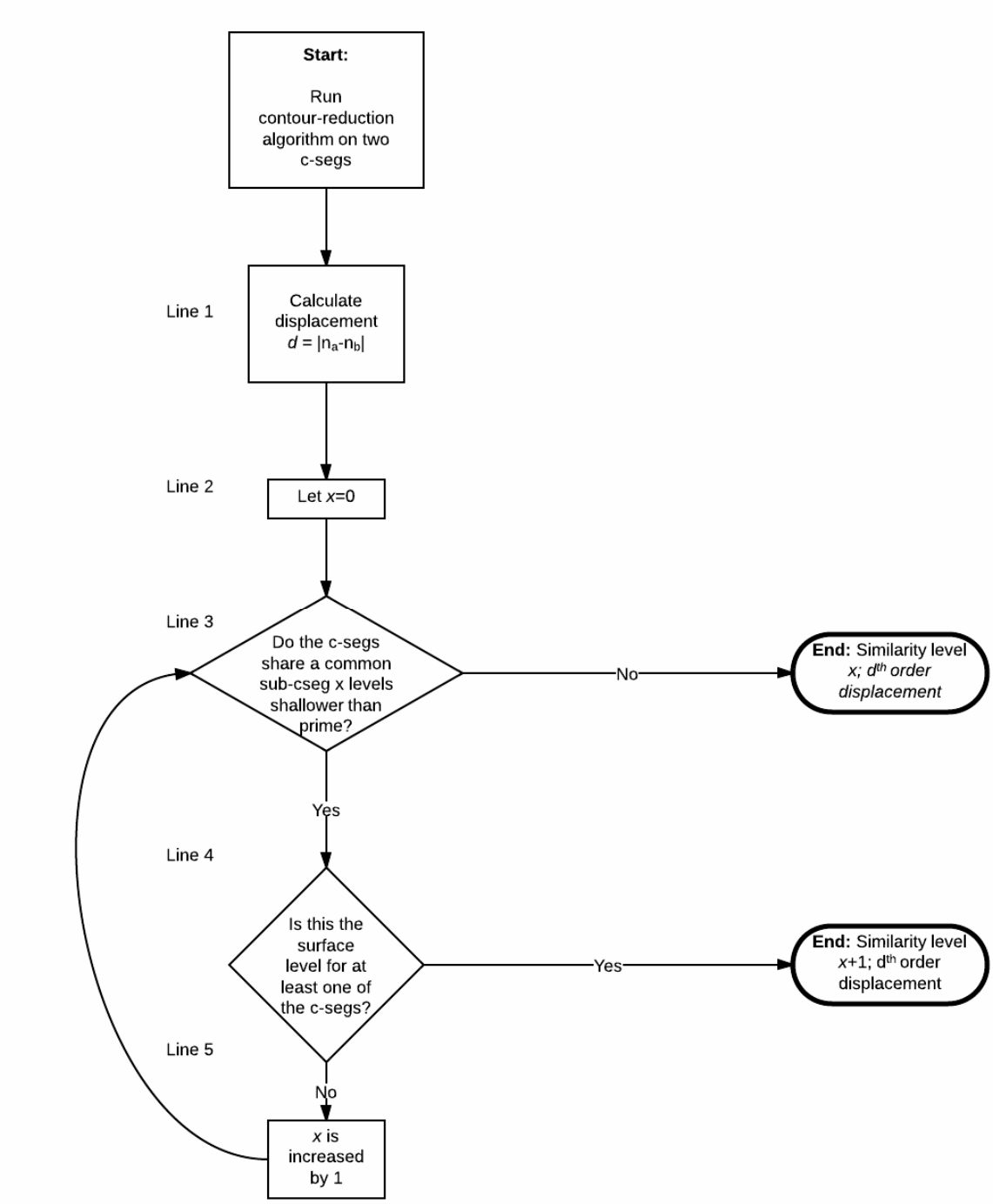


Figure 3.5 presents a modification of the comparison process for c-segs that possess some order of displacement. Since the comparison must proceed to examine sub-csegs regardless of this complexity difference, some way to indicate the lack of similarity on this level must exist in order to highlight the difference between a direct comparison and one occurring on some order of displacement. In order to make such a distinction clear, the complexity measurement of the direct comparison outlined in chapter 2 is replaced by the displacement calculation. For c-segs that possess some level of displacement, the displacement value *d* is calculated by subtracting the complexity of the deeper c-seg from the complexity of the shallower c-seg. Such a displacement value indicates the extent of inequality between the complexities of the two c-segs, and therefore sufficiently replaces the earlier complexity measurement.

Once the order of displacement is found, the rest of the comparison works in a manner slightly different than the direct comparison. The process, as shown in the flowchart, proceeds through the displacement calculation and sets *x* to 0. Since the sub-csegs are displaced, it is no longer possible to refer to sub-cseg levels (represented by N for each distinct sub-cseg). We must now conceive of sub-cseg comparison in line 3 of the flowchart as comparing the sub-csegs found *x* levels shallower than the prime. As such, *x* not only functions as a similarity measurement, but it also counts the number of iterations that occur at the individual depth-levels of the respective sub-csegs, regardless of whether or not their N values correspond.

Following this step on line 3 of the chart, if the two sub-csegs in question are not identical, the comparison ends. The similarity level is represented by the current value of the variable *x*, yet this similarity level label lacks crucial information regarding the

Figure 3.5. Displacement comparison flowchart



specific comparison process we used to compare these two c-segs. Currently there is no way to distinguish between a label derived by the displaced similarity comparison or the direct similarity comparison. To make this distinction, and to account for the inequality of complexity within the displaced comparison, we need to include the order of displacement, as illustrated by the end condition on line 3 of the flowchart.

If the two c-segs in question do possess an identical sub-cseg *x* levels shallower than prime, one must then move on to ask if at least one of the sub-csegs compared in the previous decision belongs to the surface level of one of the c-segs. If this is not the case, *x* is increased by 1 and the process begins again. The recursive loop repeats until one of the end conditions is met, at which point the displacement is taken into account and the level of similarity is calculated.

With these steps, we may now more accurately describe the similarity between the c-segs presented in Figure 3.3. The displacement *d* is calculated by subtracting the complexity of c-seg *a* (N_a=3) from the complexity of c-seg *b* (N_b=2) to arrive at a displacement value of 1. We then enter line 3, where we find that both c-segs share the same prime and thus meet the criteria for level 1 similarity. Since neither of these are a surface-level c-seg, *x* is increased to 1 and sub-csegs one level shallower than prime are compared. Since the two c-segs share a common sub-cseg at this level (sub-cseg ⟨2010⟩), they meet the new criteria for level 2 similarity. Neither of the ⟨2010⟩ sub-csegs represents a surface level c-seg, so the recursion repeats again: *x* becomes 2 and the sub-csegs two levels shallower than prime are compared. They do not share a common sub-cseg at these levels, so the process ends. The similarity level is reflected by *x*, so the two c-segs share a level 2 similarity, with a first order displacement.

The Similarity Index for Displaced C-segs

Under the direct comparison, the similarity index calculates a value representing the degree of similarity between two c-segs. This index number is a crucial quantitative value that puts the distinct levels of similarity into perspective with regard to other comparisons made using this same method. As I have stated previously, the similarity index is the ratio of identical levels to the total number of comparable levels. In the direct comparison, the index relies on the fact that there are an equal number of depth levels available for comparison. However, in the displaced comparison, the depth levels do not line up in so organized a fashion.

Due to the depth level inequality within a displaced comparison, not every sub-cseg in the more complex c-seg has a corresponding sub-cseg with which to make a comparison. Once again, the displacement valued *d* counts the number of these sub-csegs in more complex c-seg that have no corresponding sub-csegs within the less complex c-seg. Consider Figure 3.3 once again. The displacement value for these two c-segs is 1, indicating that one level in c-seg *a* has no corresponding level in c-seg *b*.

The similarity index for displaced c-segs must take into account the fact that there are sub-csegs in the reduction of one of the c-segs that are not compared with any other sub-cseg. Consider Figure 3.6, which shows the comparison process of sub-csegs in the displaced comparison. C-seg *a* and c-seg *b* share a displacement of 1. Therefore, only four out of the five sub-csegs in c-seg *a* are comparable with a sub-cseg in c-seg *b*. With only four comparisons to make, the similarity index must be calculated out of only the four levels compared.

Assuming that all four of these comparisons yielded similar sub-csegs, the similarity index for the two c-segs would be 4/4, or 1.0. This may initially seem problematic, as it is clear that the two c-segs are not exactly identical, and this is where the displacement label becomes crucially important. The two hypothetical c-segs

Figure 3.6. Comparison of displaced sub-csegs

C-seg *a*	**C-seg *b***
Surface level	Surface level
Sub-cseg$_1$	Sub-cseg$_1$
Sub-cseg$_2$	Sub-cseg$_2$
Sub-cseg$_3$	Sub-cseg$_3$
Sub-cseg$_4$	

reflected in Figure 3.6 have a first order displacement, which reflects the fact that the c-segs are indeed not identical. Instead, c-seg *b* is completely embedded within c-seg *a*, with c-seg *a* having an additional level beyond the identical sub-csegs of c-seg *b*. With this label, the similarity level now reflects the high degree of similarity, while the displacement reflects the level of difference.

In the case of Figure 3.3, we can now accurately calculate the similarity index for these c-segs. C-seg *a* has four levels; while c-seg *b* has three, so only three levels are comparable. Only two of the three levels were found to be similar, therefore the similarity index in this case would be 2/3, or 0.67, with a first order displacement.

An Application of the Comparative Process

To illustrate the entire process, let us reexamine the two c-segs shown in Figure 3.2. First, each of the c-segs is run through the MCRA, as shown in Figure 3.2b.

Analyzing the sub-cseg structures, we notice that the complexities of the c-segs are unequal: d = 5-4 = 1.

Now that we have found the displacement, x is set to 0 and we may now begin examining their primes (i.e. the sub-csegs 0 levels shallower than prime). The primes are both ⟨120⟩, and neither is a surface-level c-seg, so x is increased to 1 and the process repeats. The sub-csegs one level shallower than prime are also identical (⟨1020⟩) and again, neither belong to the surface level. We repeat the recursive loop, increasing x to two, and examine sub-csegs two levels shallower than prime. The sub-cseg at this level for c-seg *a* is ⟨102120⟩ while the sub-cseg at this level for c-seg *b* is ⟨10313020⟩: clearly not identical. Since we cannot pass affirmatively through line 3 of the flowchart, we must arrive at our end condition. The c-segs in question share a level 2 similarity with a first order displacement.

Now that we have arrived at a level of similarity, we can calculate a similarity index for these two c-segs. C-seg *a* possesses six distinct levels with which to compare, while c-seg *b* has only five levels. Therefore, only five comparable levels exist. Only two of these five levels were found to be similar, so the similarity index is 0.4, with a first order displacement. Here, the prime level similarity is obscured both by the large number of levels shallower than prime that are different, as well as the fact that the primes are displaced. Such a moderate similarity index indicates that, although the two c-segs are related in that they share more than just a prime similarity, the reduction process of the surface-level features reduces cpitches at different recursive stages of the algorithm, suggesting that the phrase *hic est discipulus* (c-seg *b*) features a simpler composing out (to borrow a term from Schenker) than the phrase *dies sanctificatus* (c-seg *a*). The

reduction of certain cpitches in c-seg *b* occurs at a shallower level than in c-seg *a*, requiring c-seg *a* to continue through additional levels in order to reveal its deeper structures, and thus the similarity between them.

Conclusion

In this chapter, we have enhanced the comparative method presented in Chapter 2 by giving it the ability to compare c-segs exhibiting different complexity values. Orders of displacement have been created to allow for alternate tracks of comparison, thereby lifting the implicit restriction that only c-segs with the same complexity value could be subjected to hierarchical comparison. Such displacement values are added to the similarity labels and index in order to reflect this specific type of c-seg inequality. With this methodology in place, on can compare any two c-segs with the same prime, regardless of the primes' respective depth levels.

CHAPTER 4

MEASURING SIMILARITY WITHIN ALLELUIAS OF THE SAME MODE

Contour theory has been applied to a variety of genres in various contexts; however, one genre of music to which contour theory has not been rigorously applied is plainchant. The monophonic context of this repertoire indicates that melodic contour could be a primary method of organization. In this chapter, I will illustrate the usefulness of this analytical method in plainchant by discussing the similarities and differences between thirteen Alleluias from the *Liber Usualis* that share the same alleluia and jubilus.

Plainchant seems ideally suited for study using contour theory primarily because it exists almost entirely in the melodic domain. However, melodic contour has been primarily characterized in the literature in the simplest of terms. Alec Robertson and Abbot Ferretti (Robertson 1970, 28), Wagner (1970, 9), and Stevens (1986, page 279–80) all invoke the shape of an arch, but an "arch" is not always readily seen or heard on the surface of the music, which is quite often teeming with local changes in direction that mask such underlying structures. Melismatic chants, for example, contain many embellishments that can considerably obscure the overall arch shape. Syllabic chants on the other hand may exhibit the arch shape on a more visible and aurally perceptible level.

Figure 4.1a reproduces the antiphon Robertson uses in his example. Robertson states that "this 'arch' may be seen…rising from *videntes* to the point of climax (*domum)*

Figure 4.1a. *Videntes stellam* (Robertson 28)

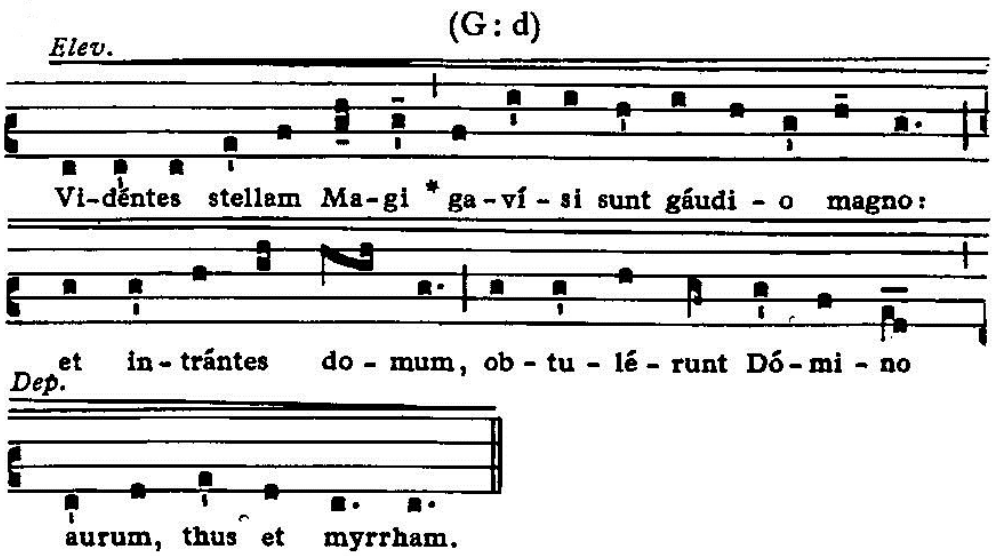
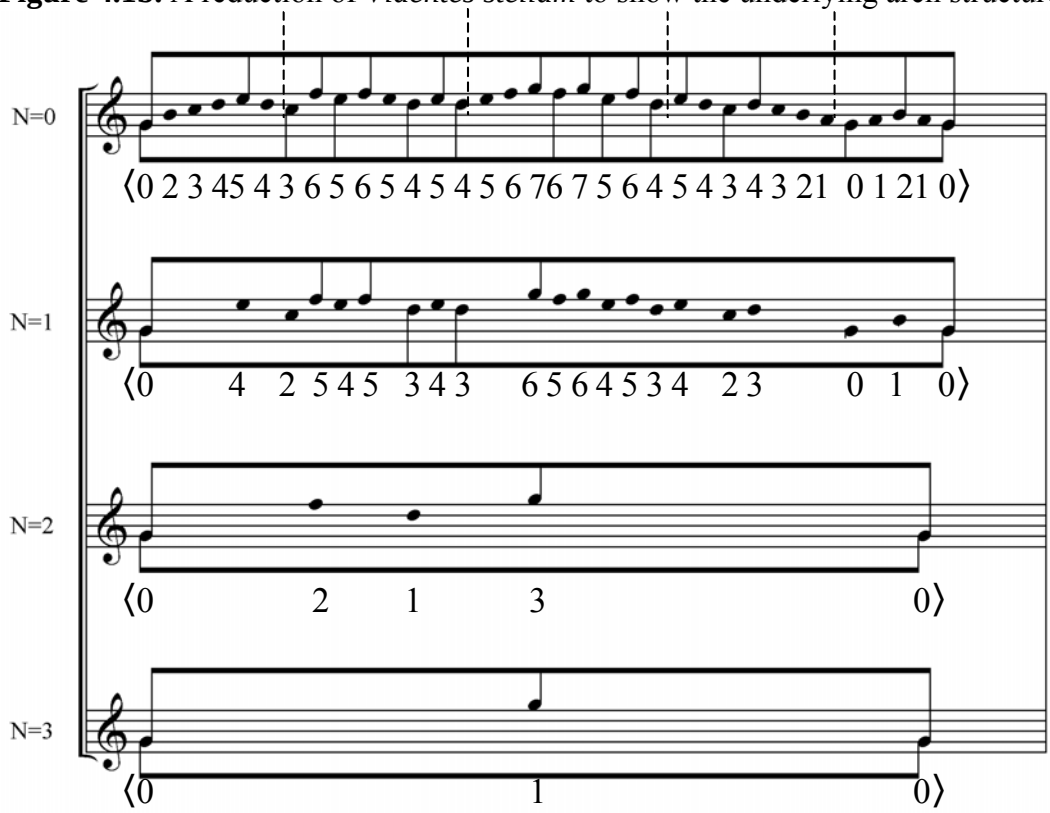
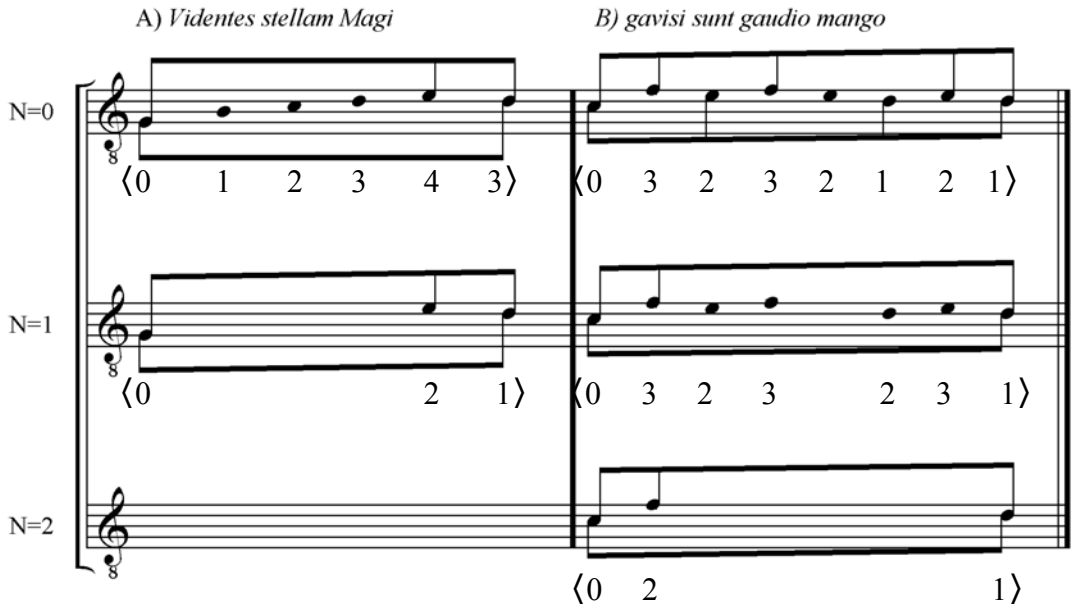


Figure 4.1b. A reduction of *Videntes stellam* to show the underlying arch structure



and falling to its other base (*myrrham)*" (28). On the deepest level, this is true; the chant does reduce to ⟨010⟩ at depth level 3, as shown in Figure 4.1b. However, this prime label does not tell us much about the structure of the music closer to the surface. Even in as simple a chant as this, interesting parallels occur both on shallower levels of reduction and within smaller segmentations. Each phrase also has an arch-like shape, and the parallelisms between the parts that make up the whole of the antiphon inform the manner in which the overall arch is formed. The first phrase, *videntes stellam magi*, features a ⟨021⟩ prime at depth level 1, and the next phrase, *gavisi sunt gaudio magno*, is also ⟨021⟩, this time at a deeper level of 2. As shown in Figure 4.1c, both are arch-like, yet neither returns to 0, giving the rise in pitch needed at the beginning of the larger overall arch. The middle phrase, *et intrantes domum*, features a prime of ⟨010⟩, and contains the climax of the entire chant on the first syllable of *domum*. The prime of the penultimate phrase, *obtulerunt Domino*, is ⟨120⟩, the inverse of the opening two primes. This helps produce the descent needed to return and complete the larger arch shape. This ⟨120⟩ also shares a level 2 inversional similarity with the ⟨021⟩ of the second phrase (*gavisi sunt gaudio magno*)—that is, their sub-cseg_2s are also related by inversion—lending further credence to the close connection between them. Finally the ⟨010⟩ of the final phrase, *aurum, thus et myrrham*, both completes and parallels the larger arch structure of the overall chant. These various phrase-level features closer to the musical surface are important to the sense of melodic unity within the chant.

Figure 4.1c. Reductions of phrase 1 and phrase 2 of *Videntes stellam*



Of course, this analysis calls into question what exactly is meant by the term "arch." Strictly speaking, an arch shape would start on a pitch, rise to a height, and return to the starting pitch. This interpretation would leave room for only one possible prime, ⟨010⟩, to govern all plainchants in the repertoire. This, however, is not the case, and indeed cannot be the case, given that chants in certain modes cannot fit this condition.[14] Wagner thus defines an arch somewhat more precisely: "as a rule the melodic line begins at a low pitch, rises to a point of climax and gradually descends to its final" (1911, III.9; quoted in Stevens 1986, 279–80). Wagner's rule expands the strict definition of "arch" to

[14] Hucbald (c. 870) discusses the relationship between the final and the starting pitch. Ostensibly, the final of the mode shall also be the final note of the chant, as is the case in the antiphon described above. However, Hucbald also states: "the four finals also possess somewhat of a like-relationship to the notes a fourth below, and in certain cases a fifth below, but such notes are used for beginnings, not endings" (39). This suggests the possibility of beginning a chant on a note other than the final, as many do. This would especially be true of the plagal modes, as more of their range extends below the final. To exclude those pitches from a chant not only would severely limit the available range of a chant, it would also obscure the plagal identity of the mode in question.

include any prime c-seg that has a prominent high point, but no unique low point outside of the first or last c-pitch. The list of possible primes thus also expands to include ⟨021⟩ and ⟨120⟩, i.e. all c-segs that belong to the CAS ⟨+, -⟩. However, many other chants feature prime c-segs other than these arch-shaped primes, even on the global scale. It also fails to explain the intricacies of the contours formed by more complex, melismatic plainchants, such as the Alleluia.

Consider, for example, the chant *Alleluia Angelus Domini Descendit,* shown in Figure 4.2a. The overall arch shape, as defined above, does not occur in this chant: instead of the ⟨+, –⟩ provided by the simpler arch structures, it exhibits a prime of ⟨10201⟩ with a CAS of ⟨–, +, –, +⟩, which is somewhat further removed from the arch idea. In this case, the meaning behind the text may have influenced shape of the various intermediary sub-csegs in order to create the ⟨10201⟩ prime. Robertson's (1970, 88) analysis of the chant discusses the manifestations of "descent" within the melodic characteristics of the chant. The text reads "Alleluia. The angel of the Lord came down from heaven: and approaching he rolled back the stone, and sat upon it." The descent of the angel in the text could be responsible for the downward interjections transforming the arch-shape on the prime level. The prime ⟨1021⟩ of the alleluia and jubilus (Figure 4.2b) begins with a prominent descent—the motion from 1 to 0—thus transforming what otherwise would be a classic arch shape. Only the verse's ⟨021⟩ prime presents an arch shape with no interrupting descents on the prime level (Figure 4.2c). However, the intermediary sub-csegs do feature prominent descents, as Robertson claims. In addition, when taken with the alleluia and jubilus, the ⟨021⟩ of the verse is further obscured. The entire chant exhibits a ⟨10201⟩ prime, which obscures the arch-shape of the verse with

two interrupting descents (Figure 4.2d). In such an alleluia, a more nuanced understanding of the contour of the chant on various levels can yield new insight into the unique structural qualities of the particular chant.

Figure 4.2a. *Alleluia Angelus Domini Descendit*

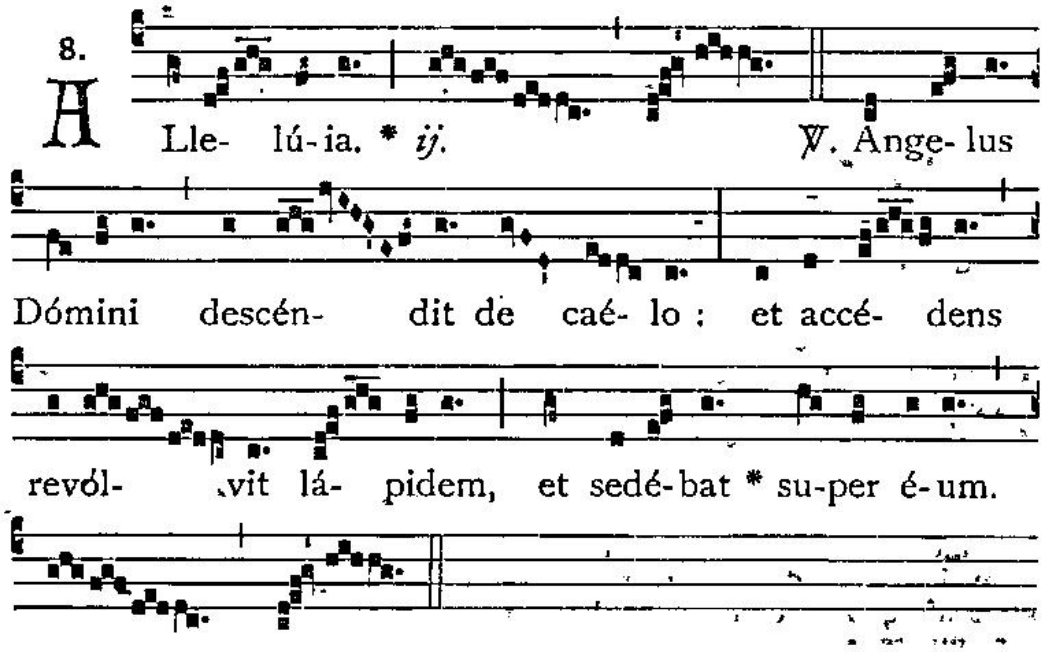


Figure 4.2b. The alleluia and jubilus from *Alleluia Angelus Domini Descendit*

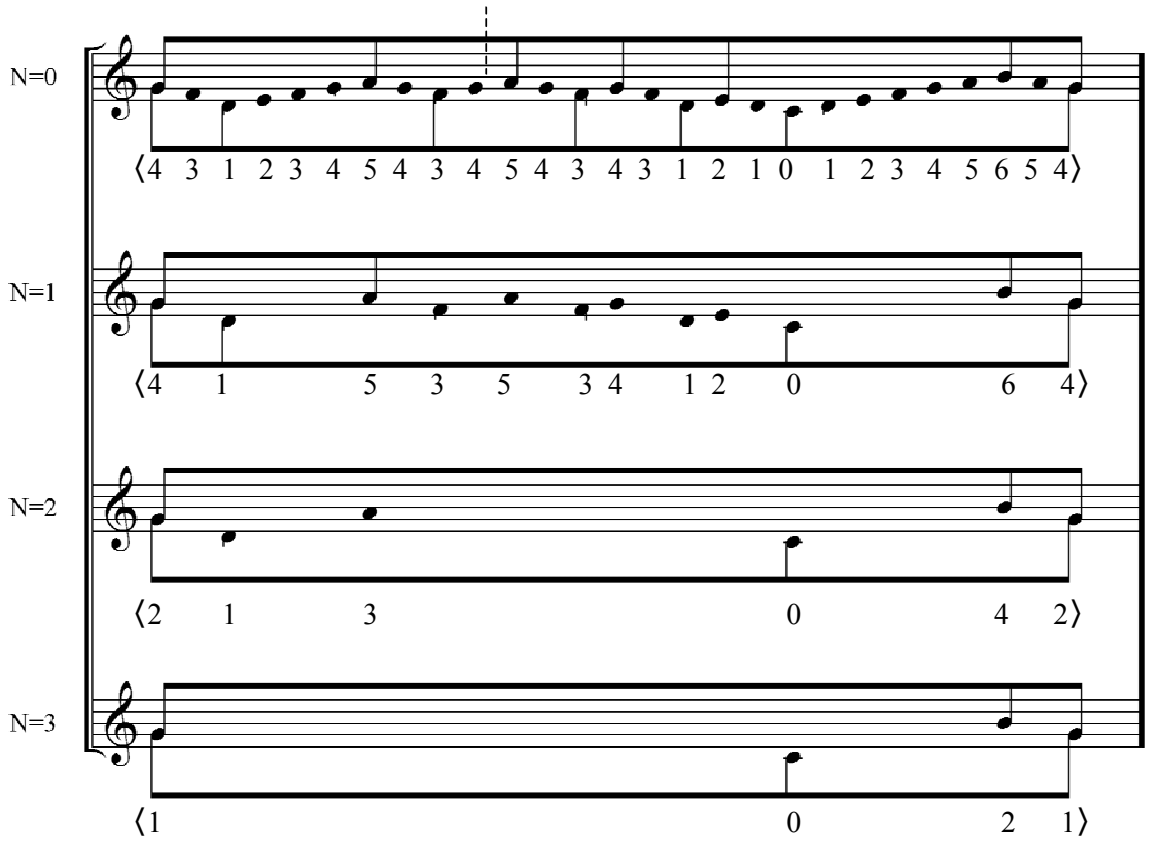


Figure 4.2c. The verse from *Alleluia Angelus Domini Descendit*

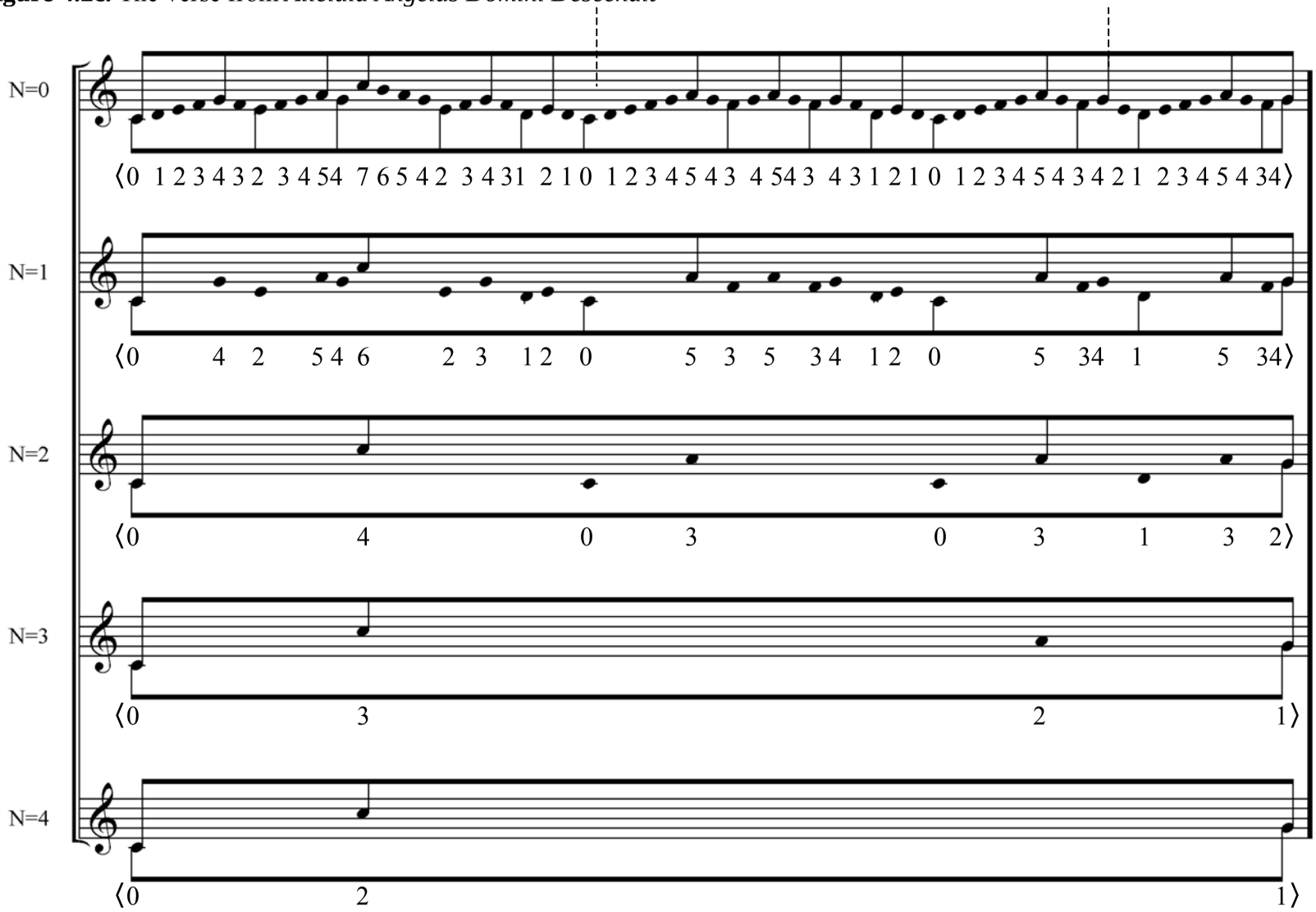
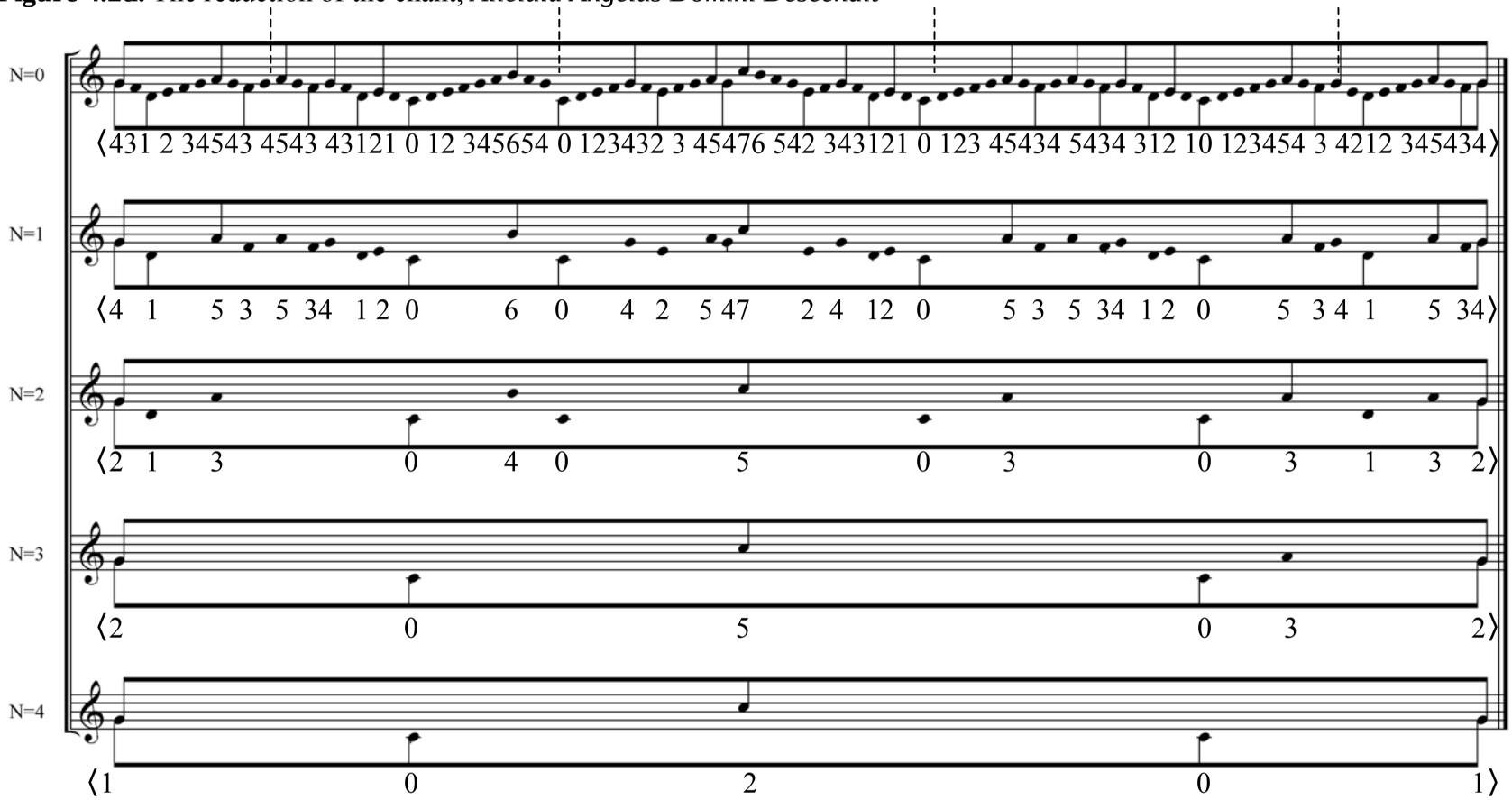


Figure 4.2d. The reduction of the chant, *Alleluia Angelus Domini Descendit*



Applying the Hierarchical Comparison Method

The MCRA and the hierarchical comparison that follows can be applied in multiple ways to contribute to our understanding of the structure of Alleluias, and more abstractly, plainchant in general. Recall that the MCRA prunes c-pitches in order to reach a prime at the deepest level of the music, and that this prime will consist of the first, last, highest, and lowest c-pitches in the c-seg. If one is looking to find the prime of a plainchant melody, several factors influence the possible outcomes of the algorithm. According to the author of the *Dialogus de musica*, the final note determines the modality; therefore the identity of the final pitch is always the final of the mode.[15] This may not seem to explain anything about contour or the possible prime outcomes of the MCRA, but due to modal structure, the fact that the final pitch is fixed becomes very important. As Wagner states, a chant will typically start on a low pitch, rise to a height, and then descend to the final, implying in certain circumstances that a chant would begin on a pitch other than the final. For example, in plagal modes, the range of the modes extends down either a fourth or fifth below the final in addition to a sixth above, as shown in Figure 4.3. Indeed, it is the prominence of the lower range below the final in these melodies that gives a plagal mode its identity.

Consider Figure 4.4. Looking at the alleluia and jubilus alone, one would not be able to distinguish the plagal identity of the mode. It is not until later in the verse, where the prominent descent down to A occurs, that one can establish the plagal identity of this

[15] "'Tonus vel modus est regula, quae de omni cantu in fine diiudicat.' ('A tone or mode is a rule which classifies every melody by its final.')" (*Dialogus de musica*, quoted in Hiley 454)

Figure 4.3. Ranges of chants in the eight modes from *Dialogus de musica*

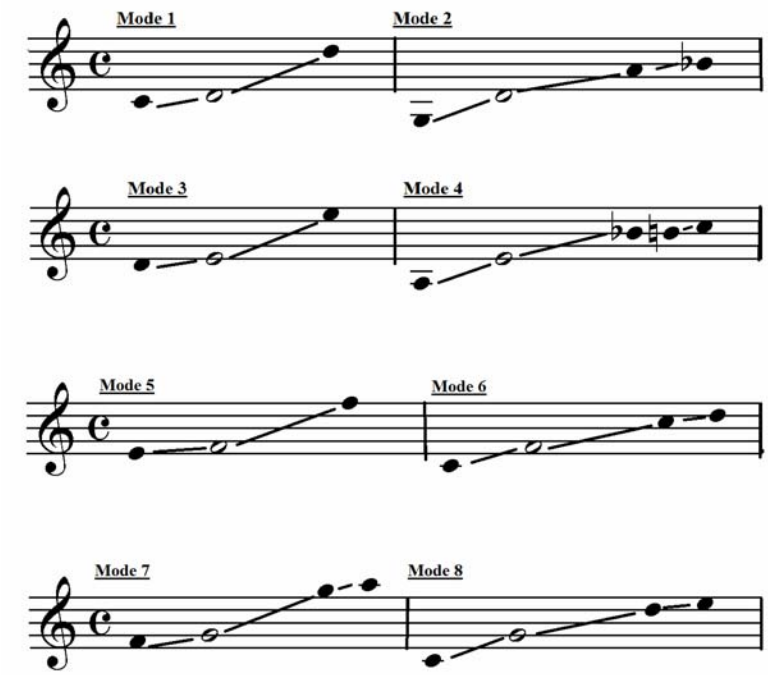
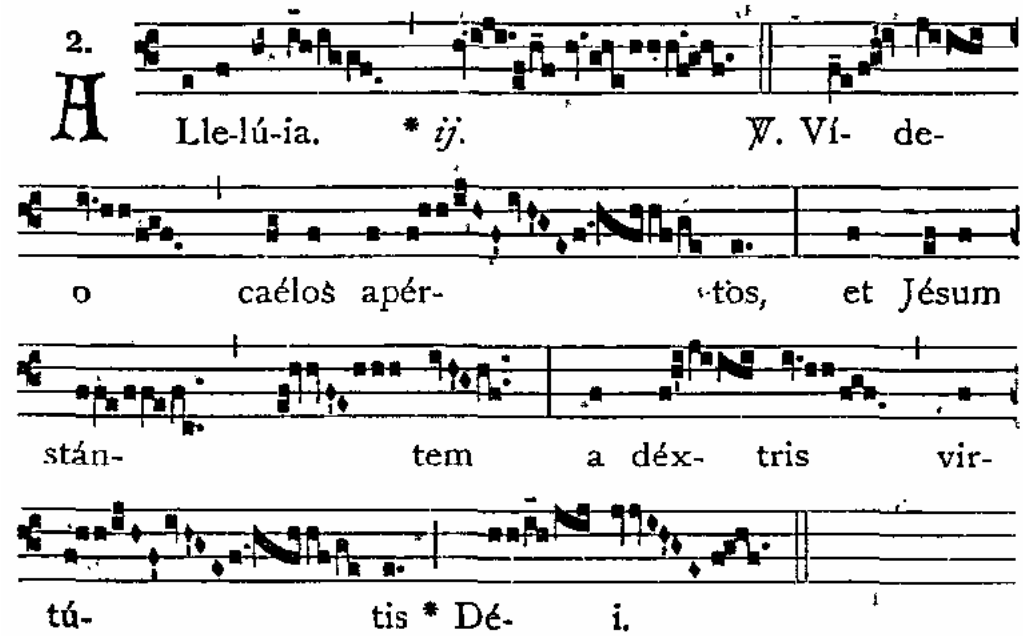


Figure 4.4. A plagal chant wherein the full range of the plagal mode is not made apparent until the middle of the chant.



chant. This descent is indeed the cause of the deviation from the overall arch-shape. With the limited range above the final, along with the melodic characteristics of plagal modes in general, it becomes impossible for the last pitch to also be the lowest. Similarly, the tendency for melodies in the authentic modes to extend a single tone below the final excludes the possibility of the last pitch being the lowest as well.

Morris (1993, 218) identifies seven linear prime classes and 22 possible linear primes.[16] Given the range requirements of these chants, five of the 22 linear primes are excluded from the possible primes for the reduction of entire alleluia chants: ⟨0⟩, ⟨00⟩, ⟨10⟩, ⟨010⟩, and ⟨120⟩. In addition, though the arch-type shape described by Robertson is perhaps too broad, the general meaning behind it remains: most chants, the Alleluias especially, begin at some point, rise up to a high point, and come back down in accordance with the CAS ⟨+,-⟩. As such, it is highly unlikely to have a prime on a global scale that begins or ends on the high point. This excludes the linear primes ⟨01⟩, ⟨101⟩, ⟨102⟩, and ⟨201⟩ as well.

With the modal and melodic constraints listed above, there are only 13 linear primes (shown in Figure 4.5) that remain as possibile large-scale primes of the Alleluias. Given that there are 170 Alleluias in the *Liber Usualis*, the fact that there are only 13 primes that could fit so many chants illustrates that there are a wide variety of possible ways in which a prime may be proliferated at shallower structural levels. For example, out of the 49 mode-1 Alleluias listed in the *Liber Usualis*, only five of the 13 possible

[16] Schultz (2008) adds two additional linear prime classes to this list, and thus 4 additional linear primes.

Figure 4.5. Table of remaining primes seen in chant

⟨021⟩
⟨1021⟩
⟨1201⟩
⟨1032⟩
⟨2301⟩
⟨1302⟩
⟨2031⟩
⟨10201⟩
⟨12021⟩
⟨10302⟩
⟨20301⟩
⟨23031⟩
⟨13032⟩

linear primes appear: ⟨10201⟩, ⟨021⟩, ⟨1201⟩, ⟨12021⟩, and ⟨010⟩. Although this suggests a high level of similarity within the generalized overall structure of a mode-1 Alleluia, this does little to reflect that these chants have varying degrees of similarity with other chants in the list. Both the *Alleluia Beautus Vir* and the *Alleluia Ego Sum Pastor Bonus*, for instance, have a prime of ⟨10201⟩ on the global level, (see Figures 4.6a and 4.6b), but share only level 1 similarity. Under both comparison processes, a level 1 similarity indicates a similarity between primes only: even the depth levels of these primes are different. Because only their primes are identical and there are seven levels of comparison between them, their similarity index is 0.14. This low value reflects the extent to which two chants with the same primes can differ in their melodic design closer to the musical surface.

Figure 4.6a. The reduction of *Alleluia Beatus vir*

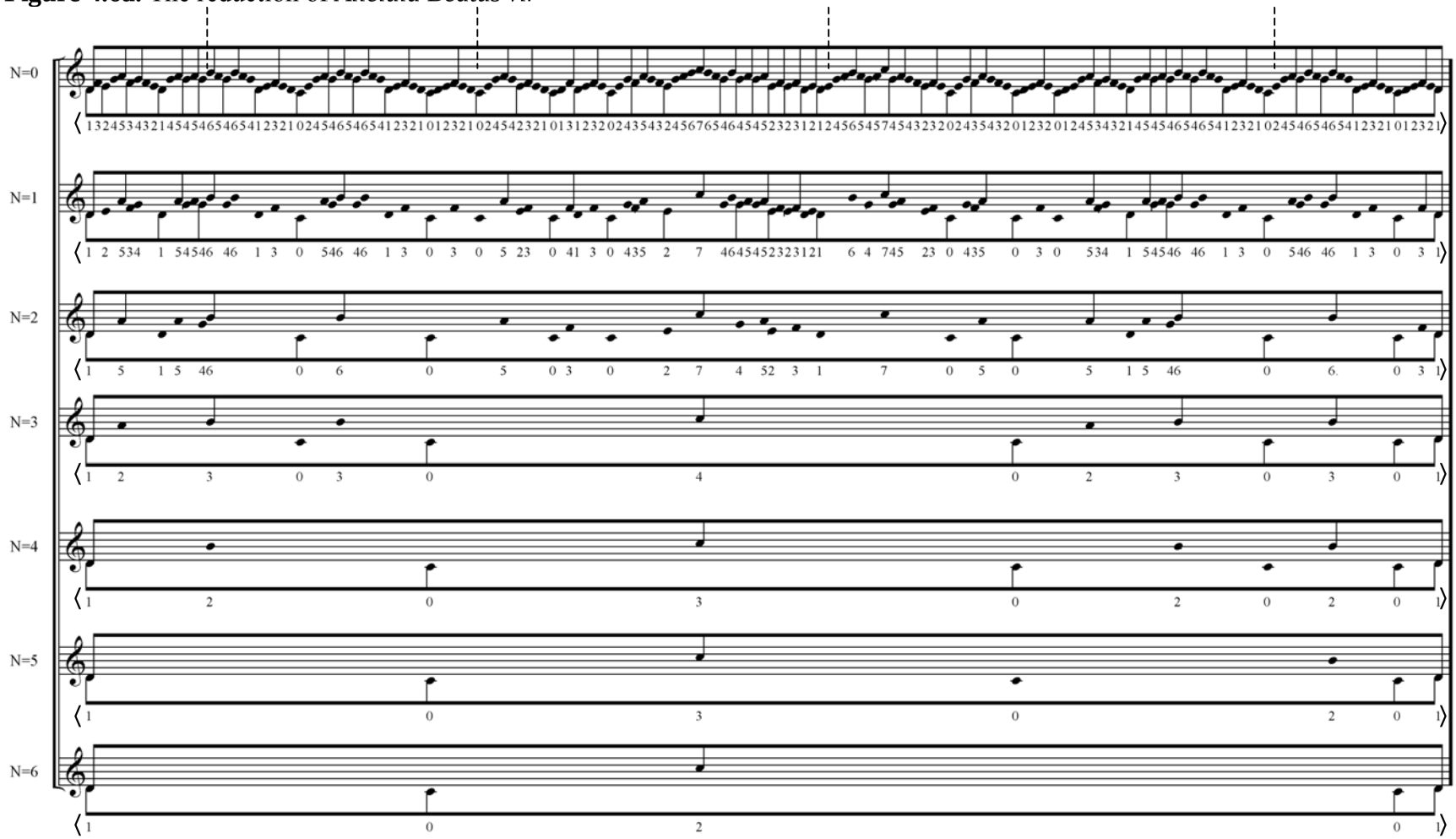
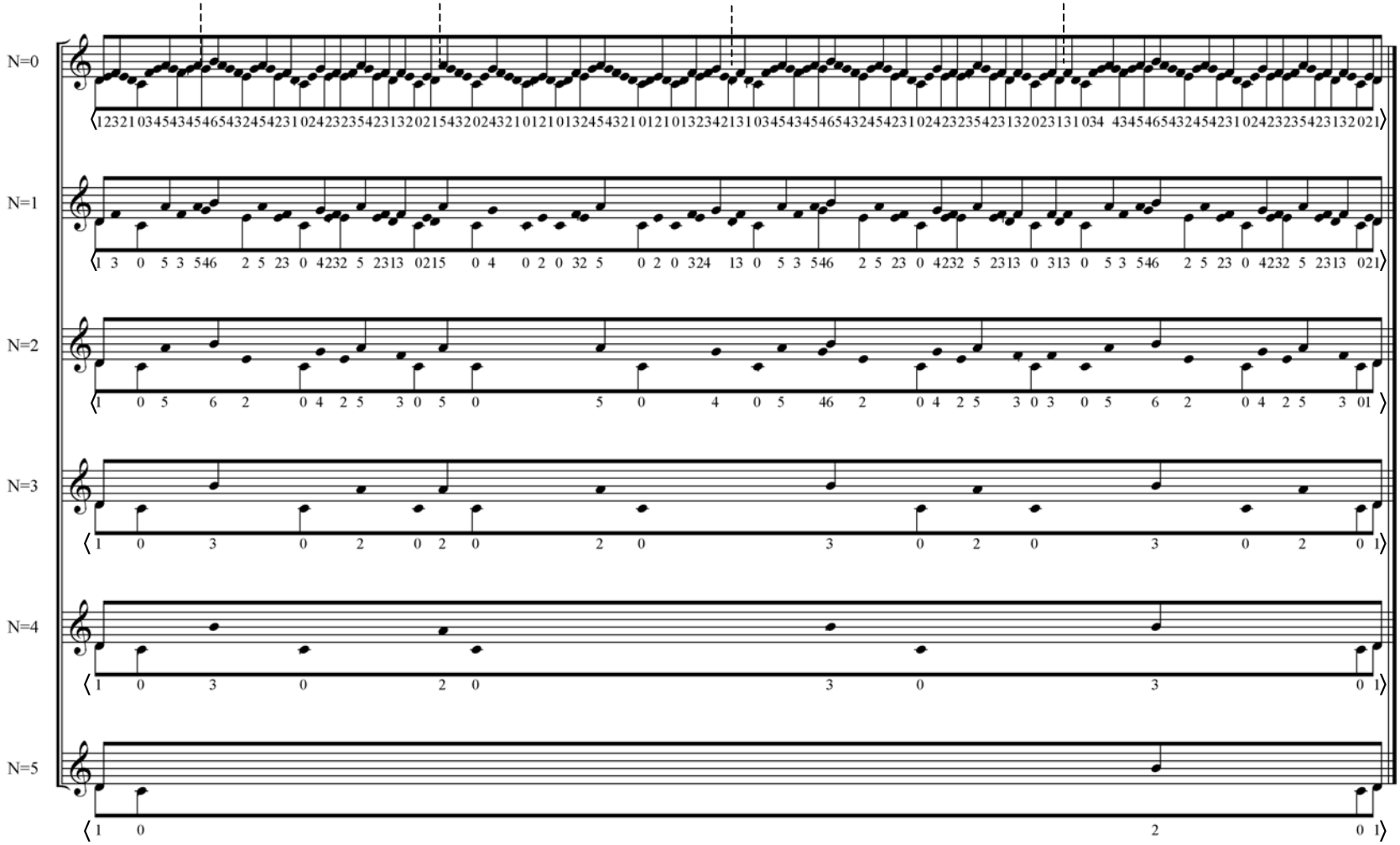


Figure 4.6b. The reduction of *Alleluia Ego sum pastor bonus*



Examining just the opening alleluia segment, which is comprised of the alleluia with the attached jubilus, provides a microcosmic example of this same phenomenon. Of the same 49 mode-1 Alleluias, only four primes occur: ⟨021⟩, ⟨1201⟩, ⟨1021⟩, and ⟨10201⟩. Figure 4.6c and Figure 4.6d shows the reduction of the alleluia and jubilus from *Alleluia Verumtamen* and *Alleluia Propitus esto*, which both have a prime of ⟨021⟩ but once again share only a level 1 similarity. As this figure illustrates, these chants exhibit a variety of differing surface-level features despite their deeper-level similarities on the prime level.

Figure 4.6c. The reduction of the alleluia and jubilus from *Alleluia Verumtamen*

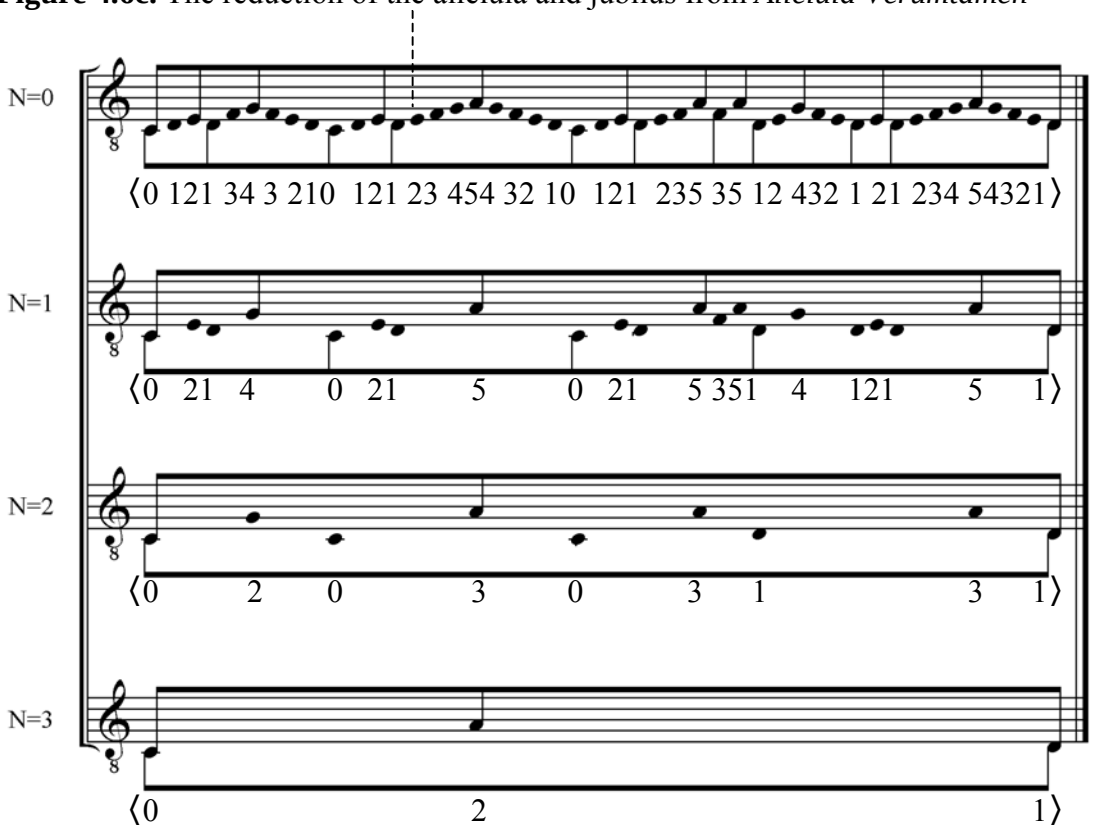
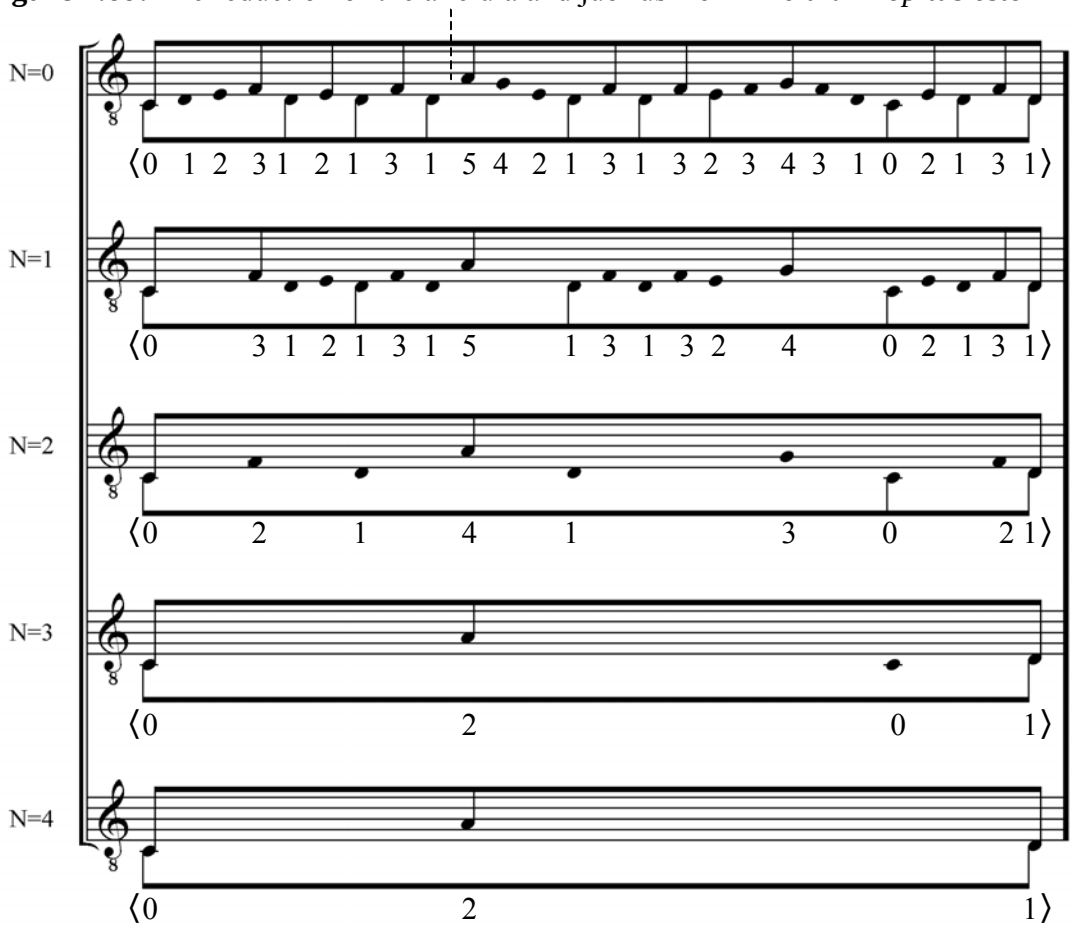


Figure 4.6d. The reduction of the alleluia and jubilus from *Alleluia Propitus esto*



Contour and "Sameness" of Melody

It is also striking that out of the 27 mode-2 Alleluias in the *Liber Usualis*, almost half of them (13) contain the exact same alleluia segment, shown in Figure 4.7a and 4.7b. Scholars such as David Hiley (1993) attribute this to the fact that the same melody was often used for many different chants.

> In the earliest books with chant texts, those edited by Hesbert (1935), there are just over 100 alleluia texts. Not all have their own unique melody, however; Schlager reckons that around sixty melodies were used (see the list in Schlager, 'Alleluia', NG). Prominent among the melodies used for more than one text are those for *Dies sanctificatus* (third Mass on Christmas Day, nine other texts in the early repertory), *Dominus dixit ad me* (first Mass on Christmas Day, eleven other texts) and *Excita Domine* (third Sunday of Advent, six other texts) (Hiley 1993, 131).

Wili Apel also notes that out of 25 mode-2 Alleluias, only 14 melodies are used (1959, 138).[17] The chant *Dies sanctificatus* is one of the 13 aforementioned mode-2 chants bearing the identical alleluia segment. According to Hiley, the nine others using this "same melody" would find themselves also among the 13 found in the *Liber Usualis*. It is clear that current research would place these 13 Alleluias under the category of "same melody," and they may indeed have originated from the same source. However, a closer examination of these 13 chants reveals that, although they are functionally very similar, each features slight differences in actual pitch content.

Let us take *Alleluia Dies sanctificatus* as a starting point, shown in Figure 4.8a. For the purposes of illustrating differences and similarities between these 13 Alleluias, I will discuss various segmentations, the reductions of which are shown in Figure 4.8b–f.[18] As with other Alleluias, the melismas contained within the chant give the c-segs at each of these segmentations a high level of complexity: their depths range from N=3 to N=5, yielding multiple intermediary levels to compare with other chants. The primes for these segmentations are shown in Figure 4.8g.

---

[17] The quantitative discrepancies between Apel and me arise from the different editions of the *Liber Usualis* in use. Apel uses the 1950 edition, whereas I am using the 1961 edition.

[18] I have segmented these phrases based on the tendencies of the modes to end phrases on certain pitches—notably the final, the *subfinalis*, and a tenor—as well as the logical division of the phrases in the text. These tend to coincide in most places with the large bar, half-bar, and *punctum-mora* (similar to the dot in our modern notation, which lengthens the affected note) symbols used by the Solesmes notation to mark phrasing for performance. I have used this segmentation for each chant consistently, allowing for direct phrase-to-phrase comparison within the analysis. Certainly other segmentations are possible, and I believe they would yield the same types of similarities and differences, provided one is consistent about them.

Figure 4.7a. The alleluia and jubilus of the 13 common mode-2 Alleluias

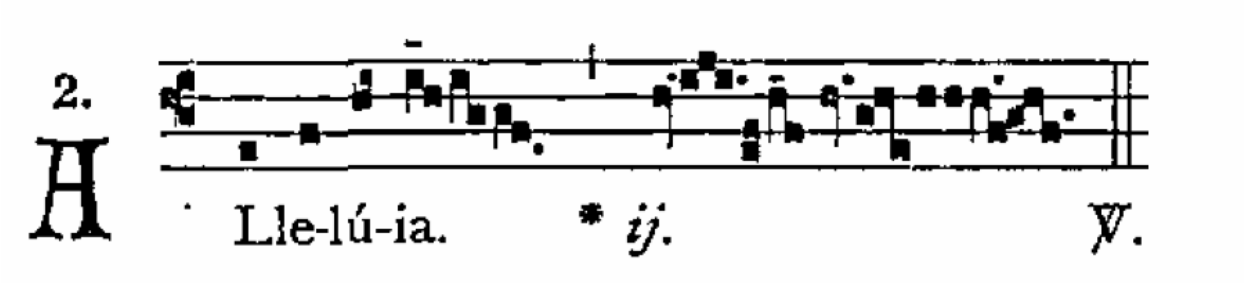


Figure 4.7b. Reduction of the alleluia and jubilus of the 13 common mode-2 Alleluias

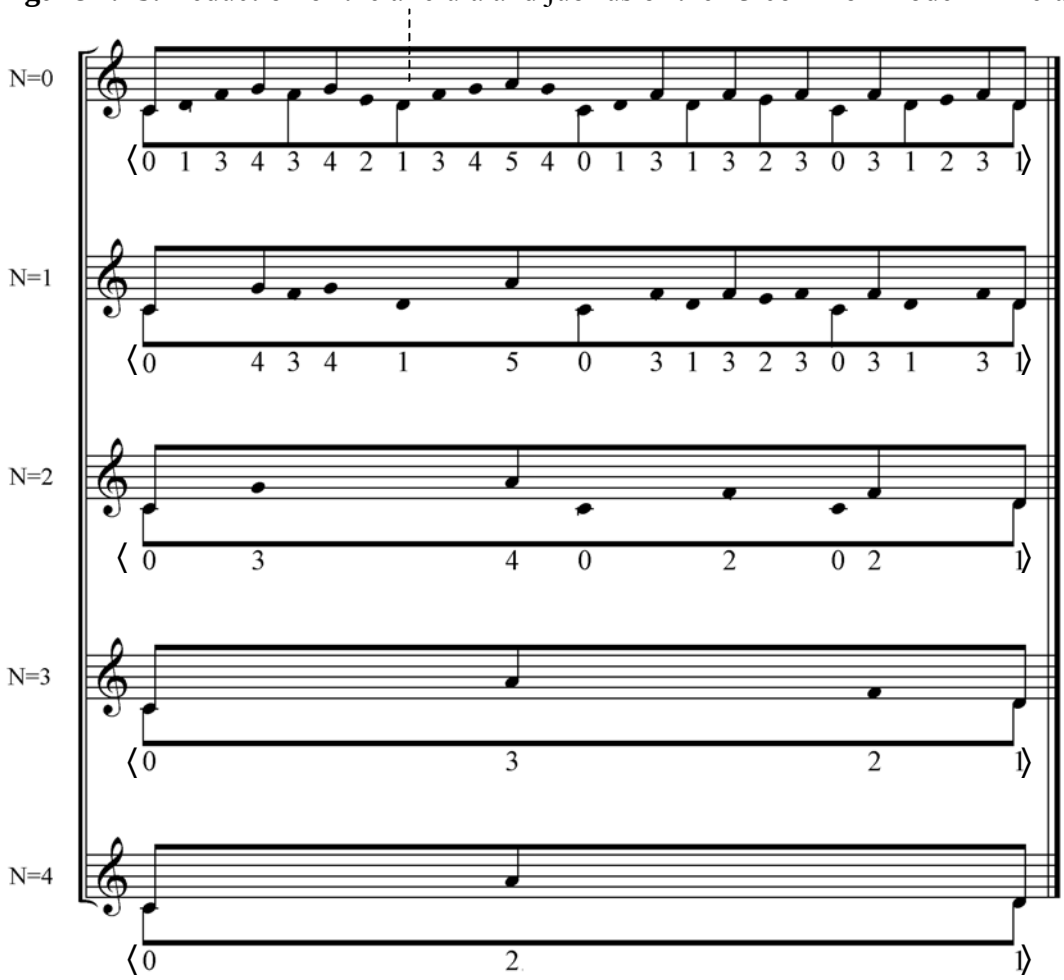


Figure 4.8a. *Alleluia Dies sanctificatus*

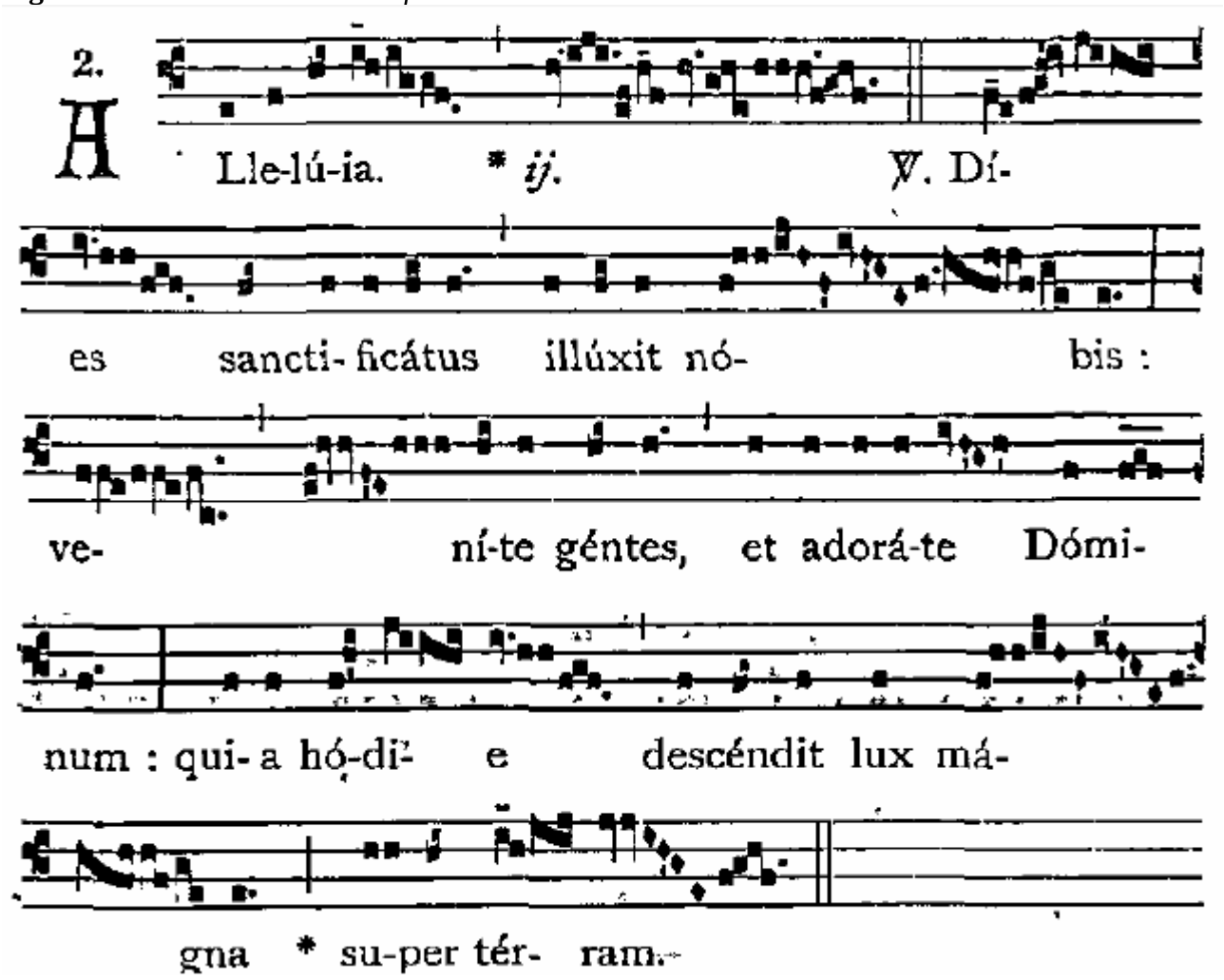


Figure 4.8b. The reduction of Alleluia Dies sanctificatus

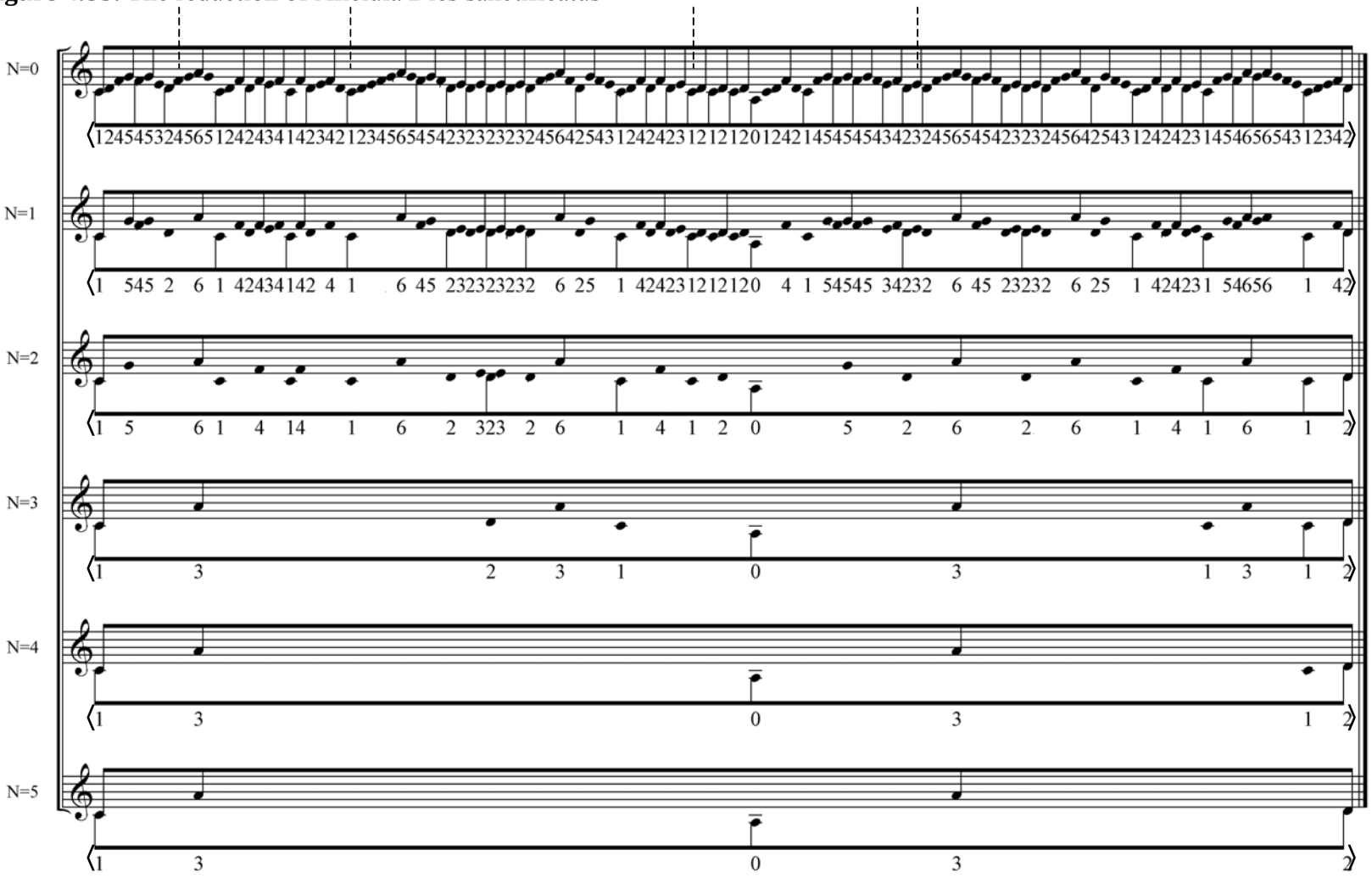


Figure 4.8c. Reduction of the verse from *Alleluia Dies sanctificatus*

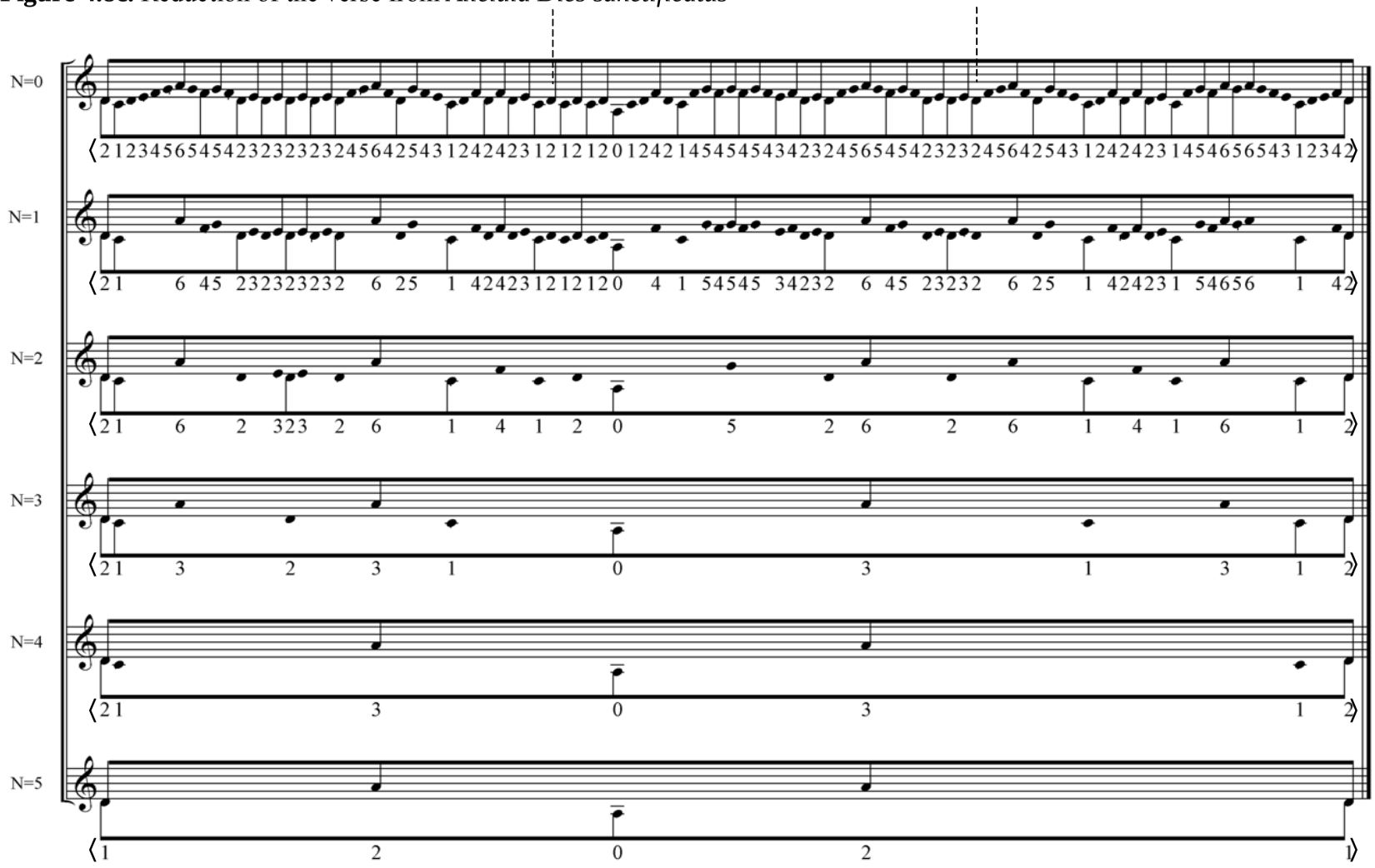


Figure 4.8d. Reduction of the phrase *Dies sanctificatus illuxit nobis* from *Alleluia Dies sanctificatus*

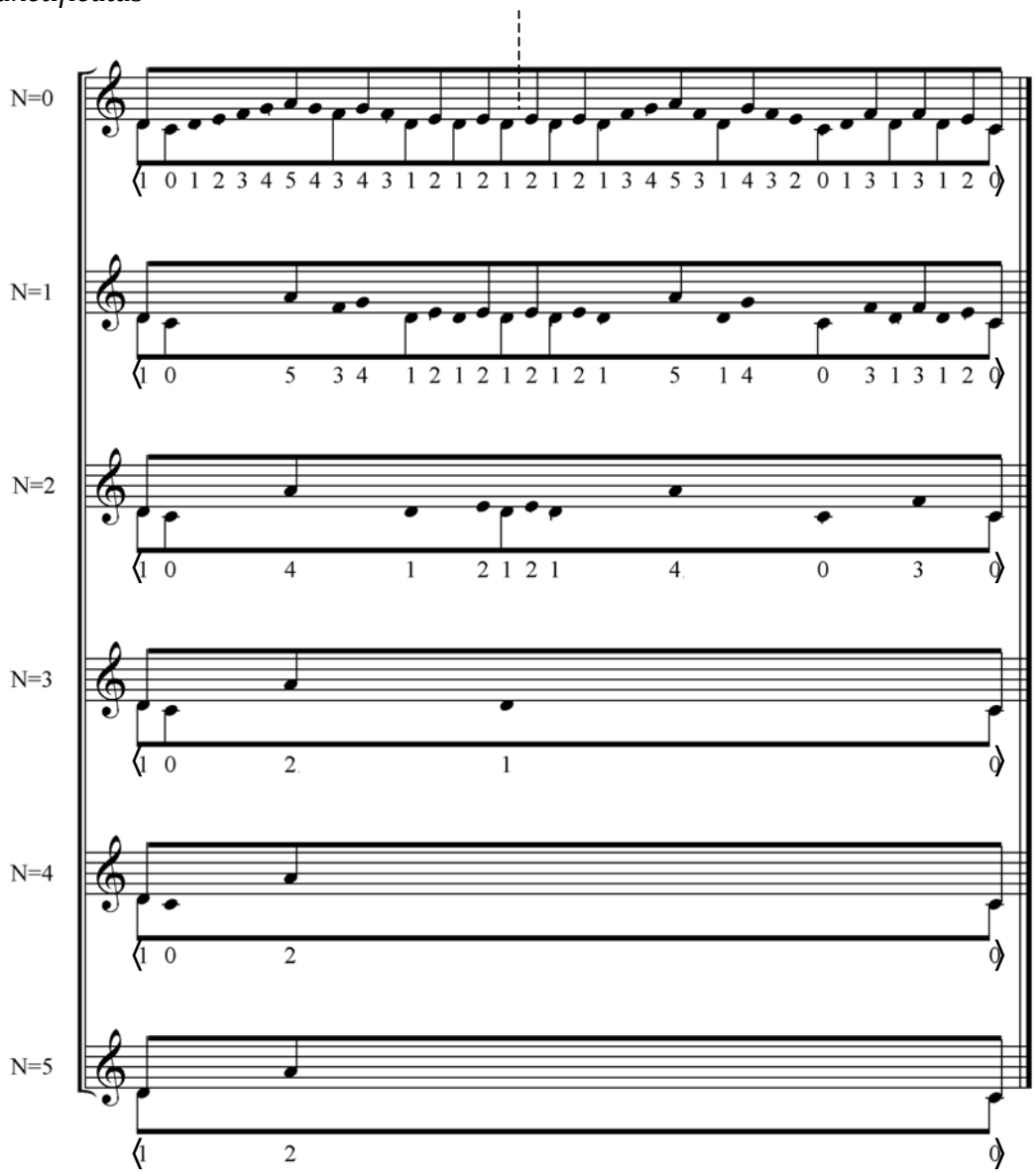


Figure 4.8e. Reduction of the phrase *venite gentes et adorate dominum* from *Alleluia Dies sanctificatus*

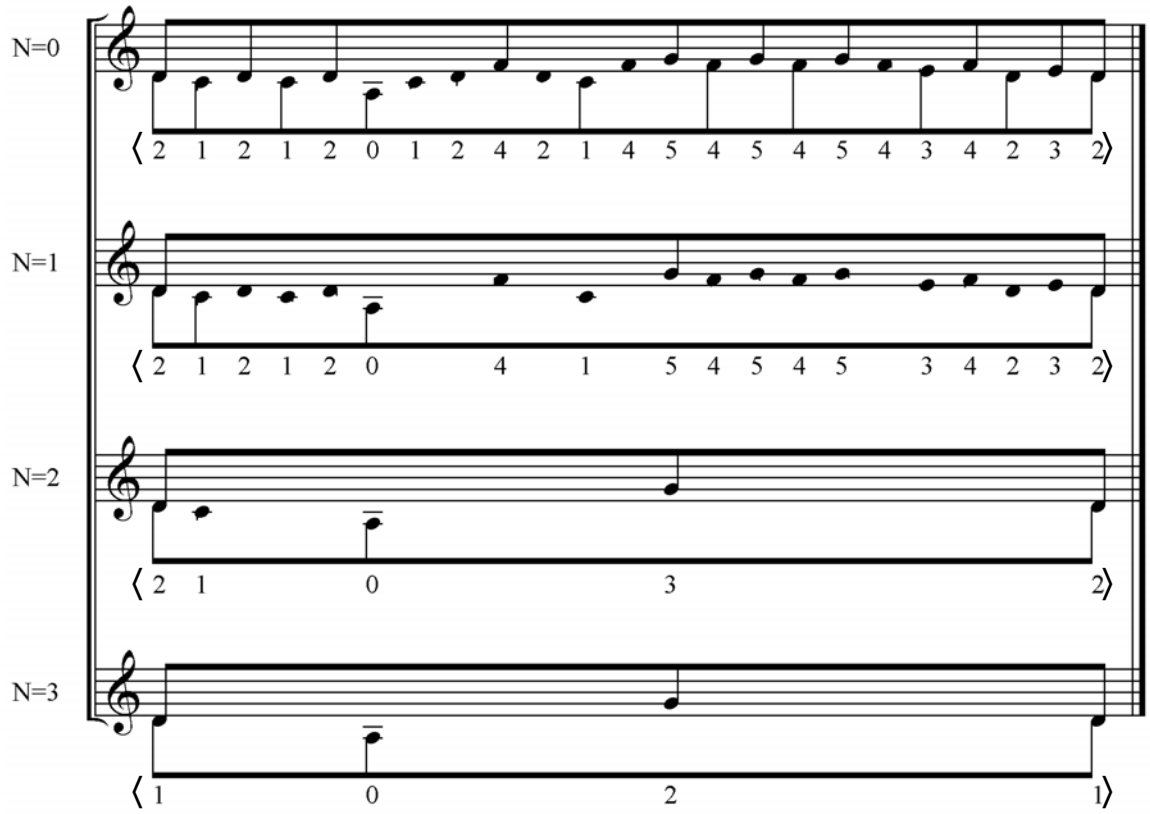


Figure 4.8f. Reduction of the phrase *quia hodie Descendit lux magna super terram* from *Alleluia Dies sanctificatus*

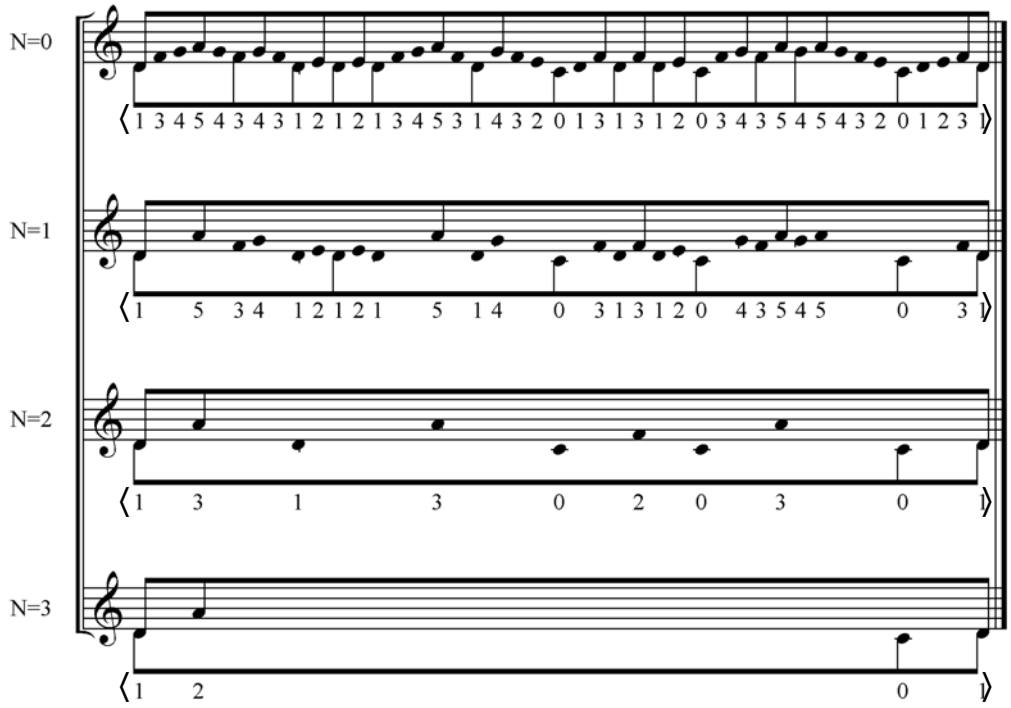


Figure 4.8g. Prime sub-csegs and depth levels of *Alleluia Dies sanctificatus*

Segmentation	**Prime**	**Depth Level**
Entire chant	⟨13032⟩	N=5
Alleluia and Jubilus	⟨021⟩	N=3
Verse	⟨12021⟩	N=5
Dies sanctificatus illuxit nobis	⟨120⟩	N=5
venite gentes et adorate dominum	⟨1021⟩	N=3
quia hodie descendit lux magna super terram	⟨1201⟩	N=3

Comparing this chant to another of the 13 Alleluias under consideration, *Alleluia Video caelos* (see Figures 4.9a–f), we see that all of the primes are identical. Comparing corresponding segments within the two chants, we find that the verses exhibit level 3 similarity, as shown in Figure 4.10, and a similarity index of 0.43. In order to pinpoint the differences that account for the remaining 0.57, we can look at the smaller-scale

segmentations. The second phrase, *Video caelos apertos,* exhibits a level 2 similarity and a first order displacement with the first chant, for a similarity index of 0.4. Both the third and fourth phrases exhibit a level 3 similarity and a similarity index of 0.6. Here, we see that all three phrases have differences, but they are not spread evenly across the verse: the third and fourth phrases have a higher similarity, suggesting that it is in the second phrase where most of the dissimilarity lies.

Figure 4.9a. *Alleluia Video caelos*

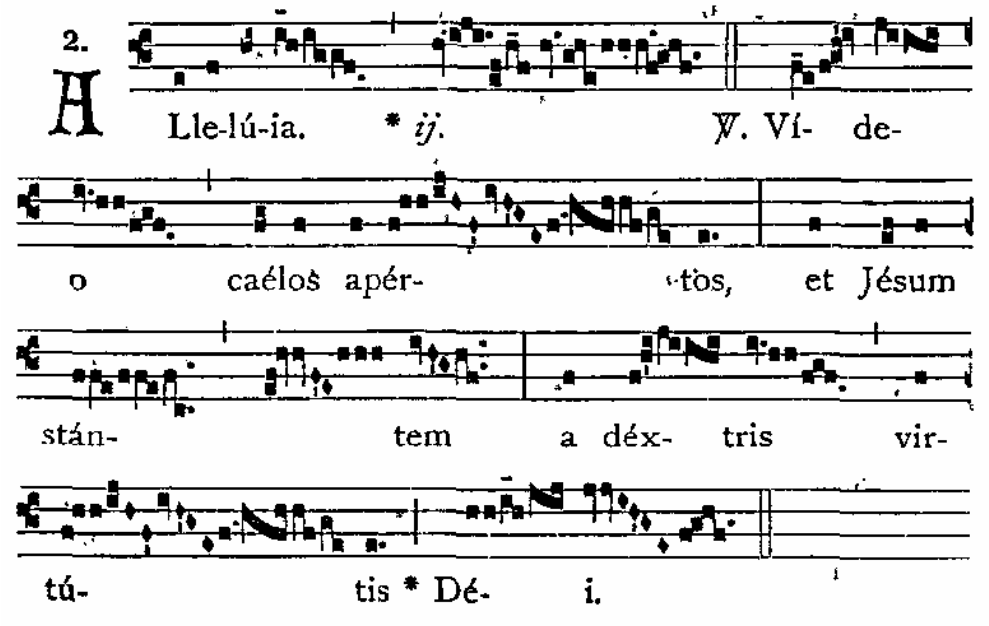


Figure 4.9b. Reduction of the chant *Alleluia Video caelos*

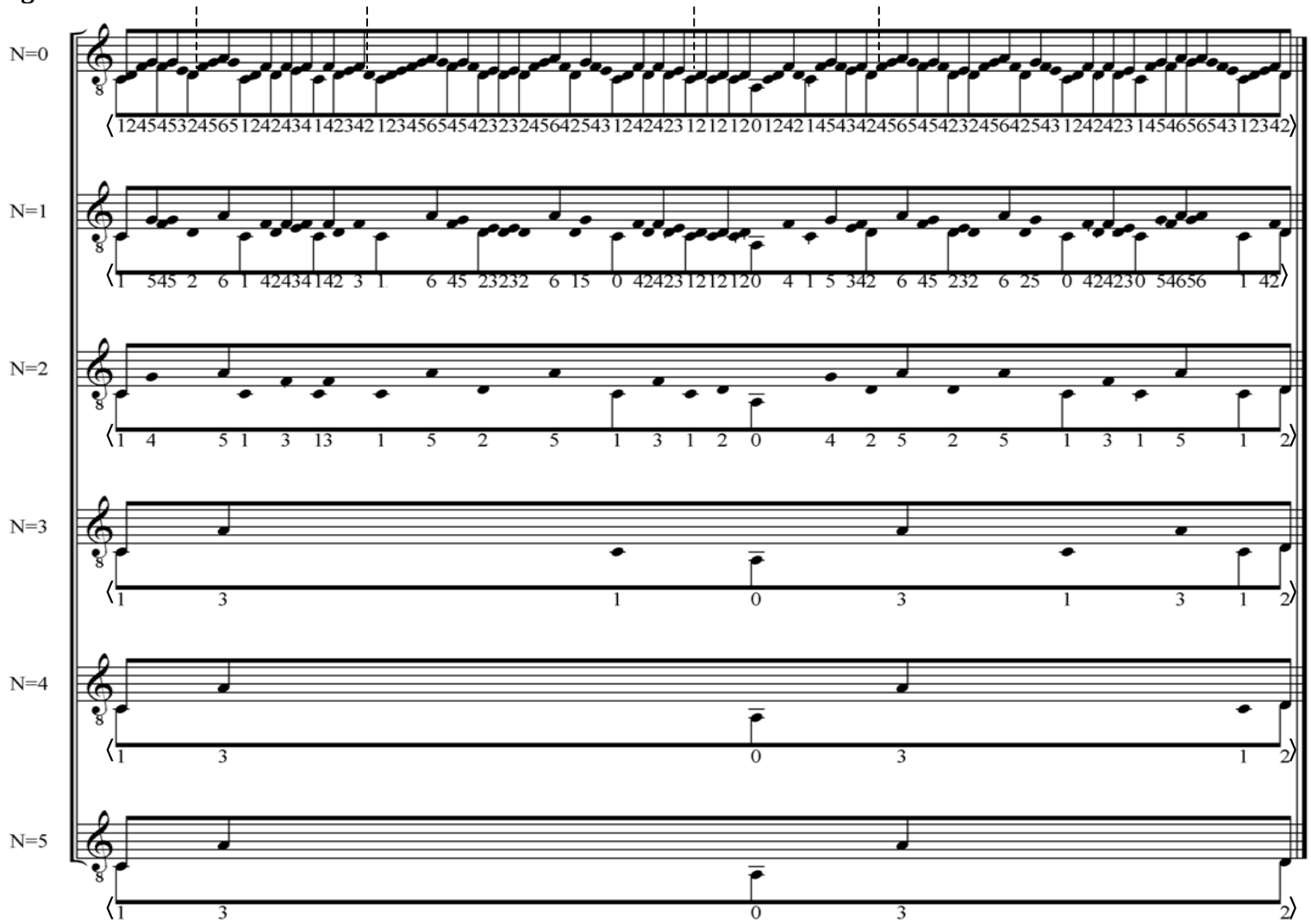


Figure 4.9c. Reduction of the verse from *Alleluia Video caelos*

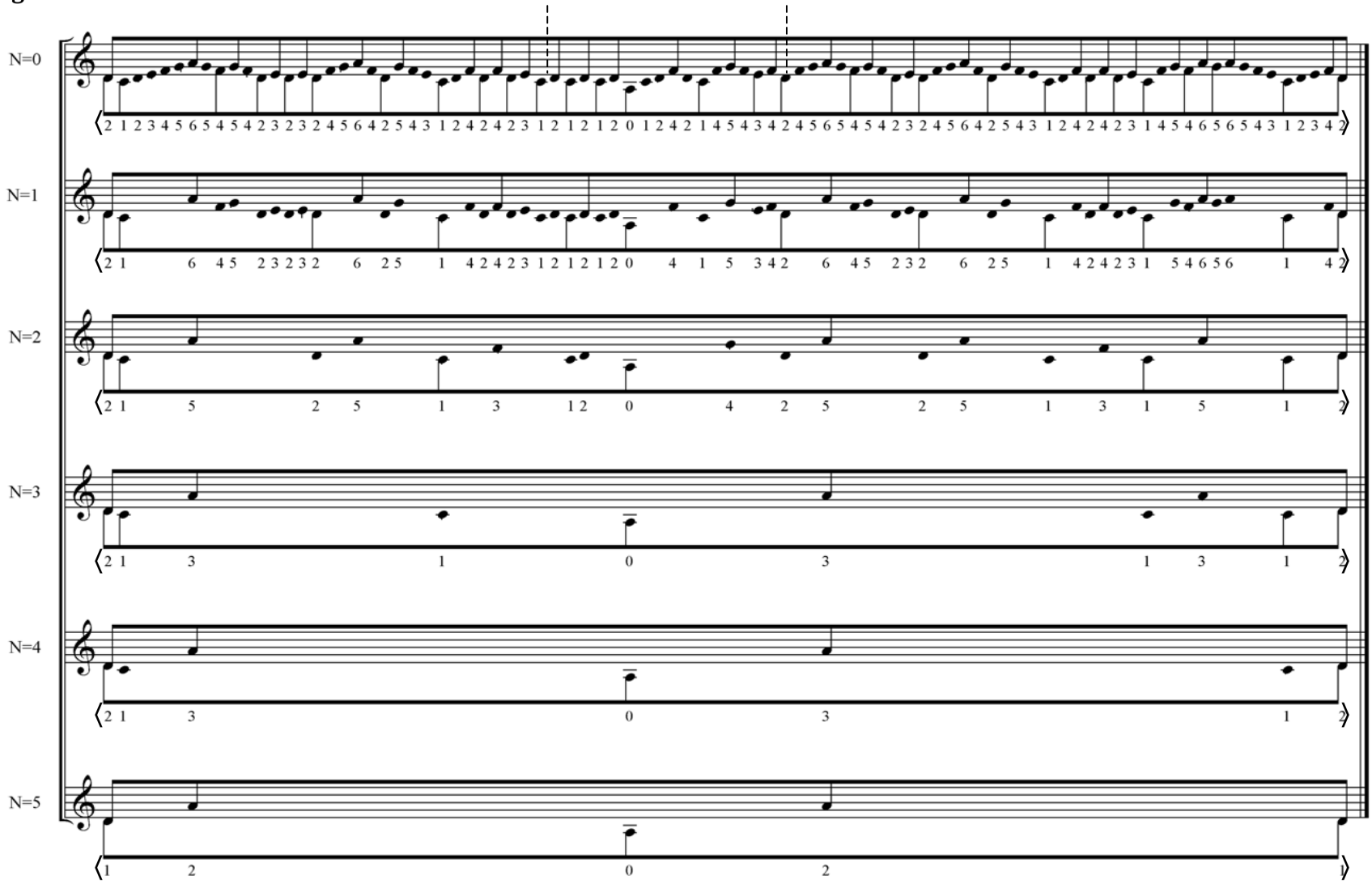


Figure 4.9d. Reduction of the phrase *Video caelos apertos* from *Alleluia Video caelos*

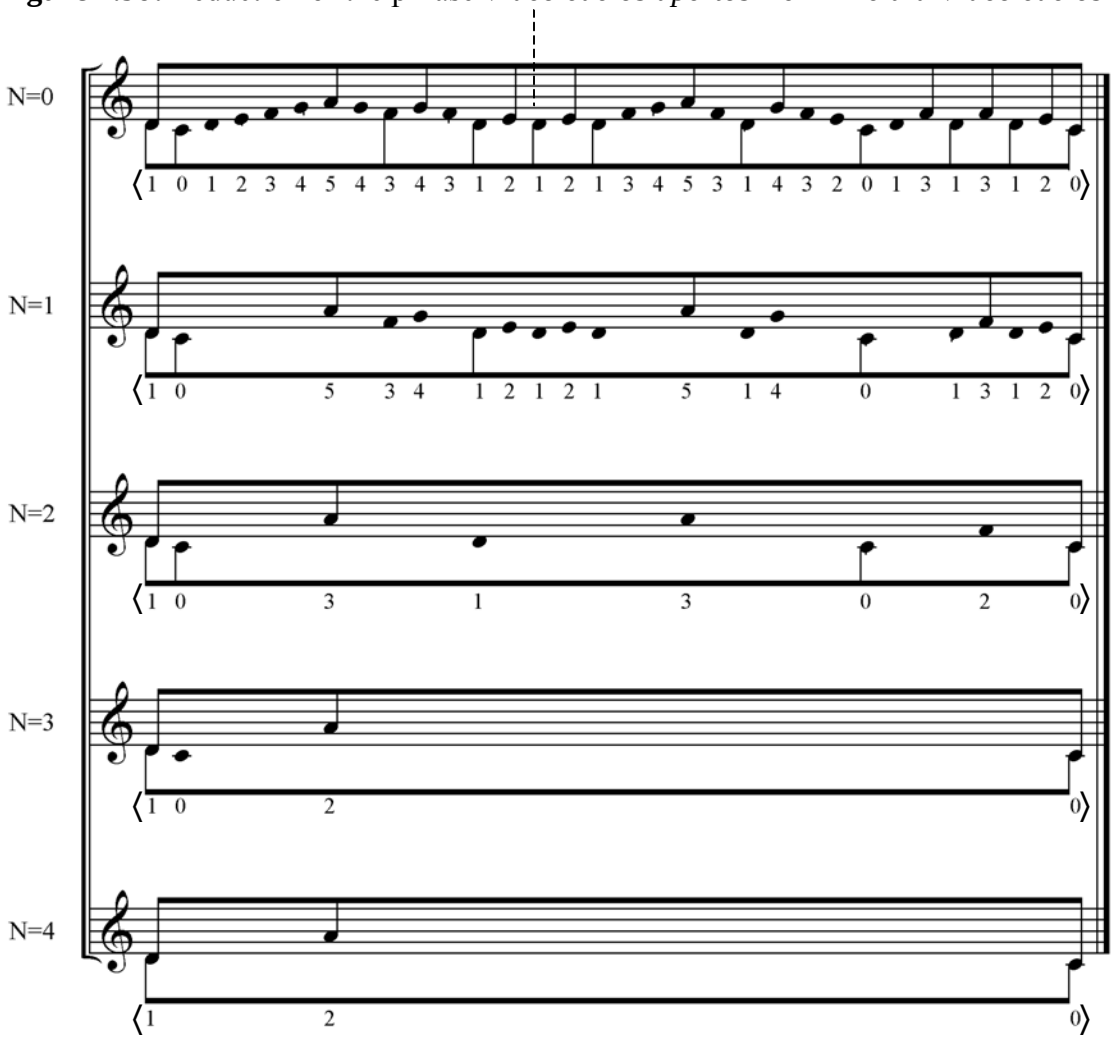


Figure 4.9e. Reduction of the phrase *et Jesum stantem* from *Alleluia Video caelos*

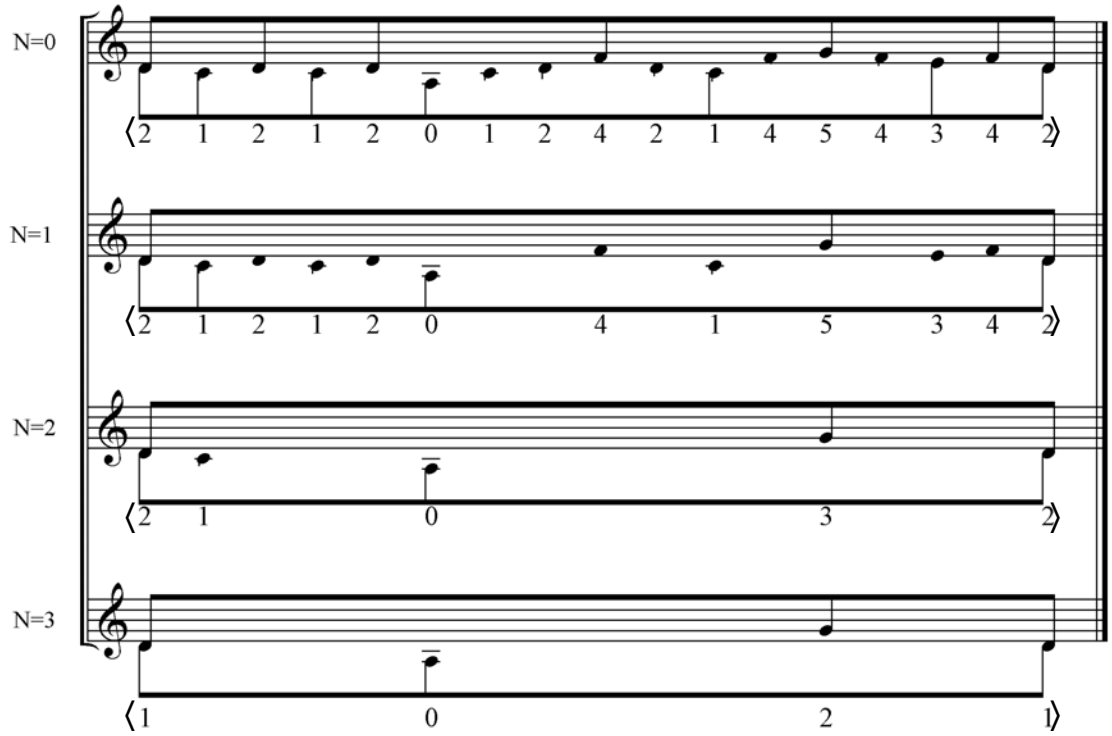


Figure 4.9f. Reduction of the phrase *a dextris virtutis Dei* from *Alleluia Video caelos*

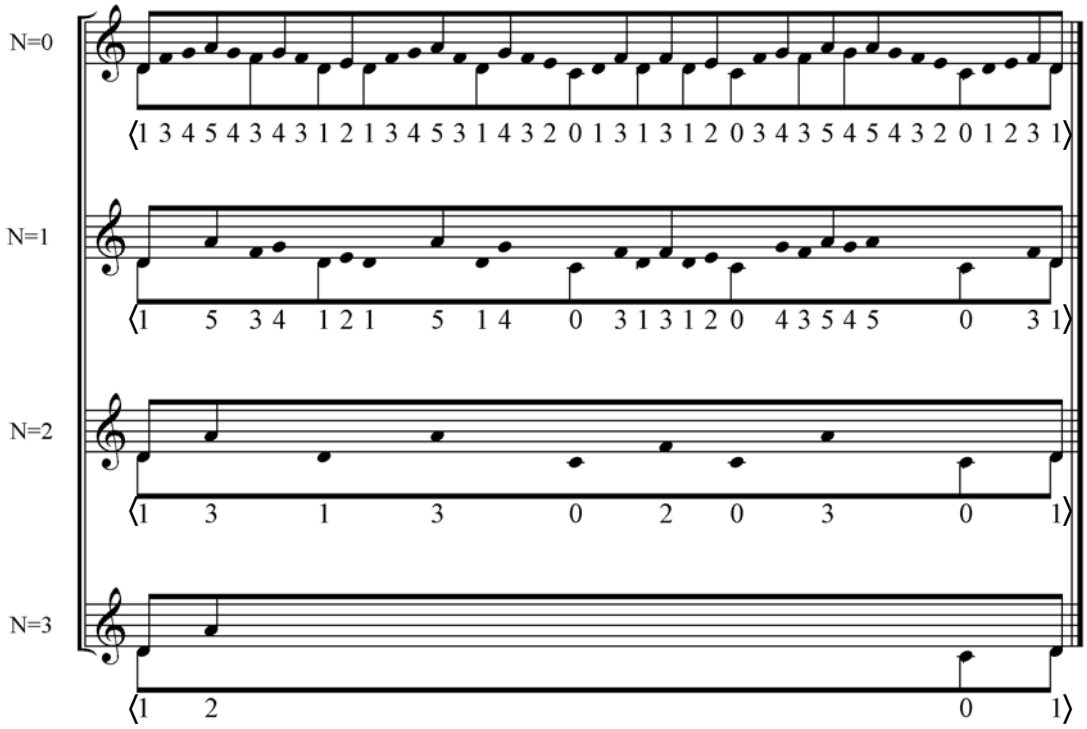


Figure 4.10. Comparison of *Alleluia Dies sanctificatus* and *Alleluia Video caelos*

Segmentation	Prime	Similarity Level	Displacement	Similarity Index
Entire chant	⟨13032⟩	3		0.43
Verse	⟨12021⟩	3		0.43
Second phrase: *Video caelos apertos*	⟨120⟩	2	1	0.4
Third phrase: *et Jesum stantem*	⟨1021⟩	3		0.6
Fourth phrase: *a dextris virtutis*	⟨1201⟩	3		0.6

An even smaller-scale segmentation reveals exactly where these differences occur. For example, the first phrase of each chant can be further divided into two parts. The first section of each—*Dies sanctificatus* (Figure 4.8d) and *Video* (Figure 4.9d) respectively—both bear the same prime of ⟨1021⟩, but share only level 1 similarity and an index value of 0.2. This reflects the major location of change for this verse segment. Turning to the second section of each verse segment—*illuxit nobis* and *caelos apertos* respectively—we see that they exhibit level 5 similarity, and are thus identical. The differences between the two phrases within these chants are therefore to be found only in their opening subphrases.

This type of analysis is not incongruent with analyses of other types of chant performed by Apel. In a discussion of the mode 2 Graduals, Apel states that "all the Graduals of mode 2 employ one and the same melody or, to put it more correctly, a small number of fixed melodic phrases that recur in various combinations" (Apel 1959, 138). Like the Gradual discussed by Apel, these small melodic units—such as the sections of the phrase—are combined, recombined, or modified to fit into a larger whole.

Figure 4.11 shows the primes and similarity levels for all the segments of the 11 remaining chants under examination, using the *Dies sanctificatus* as the point of reference.

Figure 4.11a. Similarity comparison of entire-chant c-segs with the chant *Alleluia Dies sanctificatus*

Chant name	Prime	Depth Level	Similarity level	Similarity index
Alleluia Hic est discipulus	⟨13032⟩	N=5	3	0.43
Alleluia Vidimus stellam	⟨13032⟩	N=5	3	0.43
Alleluia Redemptionem	⟨021⟩	N=4	0	0
Alleluia Tu es petrus	⟨13032⟩	N=4	1	0.14
Alleluia Hic est sacerdos	⟨021⟩	N=4	0	0
Alleluia Sancti tui domine	⟨13032⟩	N=5	3	0.43
Alleluia Magnus sanctus	⟨13032⟩	N=5	3	0.43
Alleluia Nunc com eo	⟨021⟩	N=4	0	0
Alleluia Inveni David	⟨13032⟩	N=5	3	0.43
Alleluia Tu puer propheta	⟨13032⟩	N=5	3	0.43
Alleluia Domine diligo	⟨13032⟩	N=5	3	0.43

Figure 4.11b. Similarity comparison of verse c-segs with the chant *Alleluia Dies sanctificatus*

Chant name	Prime	Depth level	Similarity level	Similarity index
Alleluia Hic est discipulus	⟨12021⟩	N=5	3	0.43
Alleluia Vidimus stellam	⟨12021⟩	N=5	3	0.43
Alleluia Redemptionem	⟨10201⟩	N=4	1 (Inversion)	0.14 (Inversion)
Alleluia Tu es petrus	⟨23031⟩	N=4	0	0
Alleluia Hic est sacerdos	⟨10201⟩	N=3	1 (Inversion)	0.14 (Inversion)
Alleluia Sancti tui domine	⟨12021⟩	N=5	3	0.43
Alleluia Magnus sanctus	⟨12021⟩	N=5	3	0.43
Alleluia Nunc com eo	⟨2301⟩	N=3	0	0
Alleluia Inveni David	⟨12021⟩	N=5	3	0.43
Alleluia Tu puer propheta	⟨12021⟩	N=5	3	0.43
Alleluia Domine diligo	⟨12021⟩	N=5	3	0.43

Figure 4.11c. Similarity comparison of first phrase c-segs with the chant *Alleluia Dies sanctificatus*

Chant name	First Phrase	Prime	Depth level	Similarity level	Displacement	Similarity Index
Alleluia Hic est discipulus	*Hic est discipulus ille:*	⟨120⟩	N=4	2	1	0.4
Alleluia Vidimus stellam	*Vidimus Stellam ejus*	⟨120⟩	N=4	2	1	0.4
Alleluia Redemptionem	*Redemptionem misit dominus:*	⟨1021⟩	N=3	0		0
Alleluia Tu es petrus	*Tu es petrus et super hanc petram*	⟨120⟩	N=3	1		0.17
Alleluia Hic est sacerdos	*Hic est sacerdos*	⟨120⟩	N=4	2	1	0.4
Alleluia Sancti tui domine	*Sancti tui domine:*	⟨120⟩	N=4	2	1	0.4
Alleluia Magnus sanctus	*Magnus sanctus paulus, vas electionis*	⟨120⟩	N=4	2	1	0.4
Alleluia Nunc com eo	*Nunc cum eo*	⟨120⟩	N=2	1		0.17
Alleluia Inveni David	*Inveni David*	⟨120⟩	N=4	2	1	0.4
Alleluia Tu puer propheta	*Tu, puer propheta*	⟨120⟩	N=4	2	1	0.4
Alleluia Domine diligo	*Domini diligo*	⟨120⟩	N=4	2	1	0.4

Figure 4.11d. Similarity comparison of second phrase chant c-segs with the chant *Alleluia Dies sanctificatus*[19]

Chant name	Second Phrase	Prime	Depth level	Similarity level	Similarity Index
Alleluia Hic est discipulus	*Qui testimonium perhibet de his:*	⟨1021⟩	N=3	3	0.6
Alleluia Vidimus stellam	*In oriente*	⟨1021⟩	N=3	3	0.6
Alleluia Redemptionem	*In populosuo:*	⟨1021⟩	N=2	1	0.2
Alleluia Tu es petrus	*Aedificabo ecclesiam meam:*	⟨2301⟩	N=3	3	0.6
Alleluia Hic est sacerdos	*Quam coronavit dominus:*	--	--	--	--
Alleluia Sancti tui domine	*Benedicent te:*	⟨1021⟩	N=3	3	0.6
Alleluia Magnus sanctus	*Verdigne est glorificandus:*	⟨1021⟩	N=3	3	0.6
Alleluia Nunc com eo	*Regnas:*	⟨120⟩	N=3	0	0
Alleluia Inveni David	*Servum meum:*	⟨1021⟩	N=3	3	0.6
Alleluia Tu puer propheta	*Altisimi vocaberis*	⟨1021⟩	N=3	5	1.0
Alleluia Domine diligo	*Habitaculum domus tuae*	⟨1021⟩	N=3	3	0.6

[19] Some of the chants are shortened, and therefore do not have either second or third verse segments. In these charts, they are left blank. This absence accounts for some of the larger scale differences.

Figure 4.11e. Similarity comparison of third phrase chant c-segs with the chant *Alleluia Dies sanctificatus*

Chant name	Third Phrase	Prime	Depth level	Similarity level	Similarity Index
Alleluia Hic est discipulus	*Et scimus quia verum est testimonium ejus*	⟨1201⟩	N=3	3	0.6
Alleluia Vidimus stellam	*Et venimus cum mune vibus adorare dominum*	⟨1201⟩	N=3	3	0.6
Alleluia Redemptionem	--	--	--	--	--
Alleluia Tu es petrus	*Ecclesiam meam*	⟨1201⟩	N=2	1	0.2
Alleluia Hic est sacerdos	*Quem coronavit Dominus*	⟨2301⟩	N=2	0	0
Alleluia Sancti tui domine	*Gloriam regni tui dicent*	⟨1201⟩	N=3	3	0.6
Alleluia Magnus sanctus	*Qui et meruit thronum duodecimum possidere*	⟨1201⟩	N=3	3	0.6
Alleluia Nunc com eo	*In aeternum*	⟨2301⟩	N=2	0	0
Alleluia Inveni David	*Oleo sancto meo unxieum*	⟨1201⟩	N=3	5	1.0
Alleluia Tu puer propheta	*Praeibis ante Dominum parare vias ejus*	⟨1201⟩	N=3	3	0.6
Alleluia Domine diligo	*Et locum tabernaculi gloriae tuae*	⟨1201⟩	N=3	5	1.0

These tables clearly indicate a degree of difference between the chant segments, as evidenced by the variety of depth levels, similarity levels, and index values. They also illustrate the need for segmenting at multiple levels. For example, the chant *Alleluia Hic est sacerdos* shares no similarity with *Dies sanctificatus* on the global level (Figure 4.10a). However, closer examination of smaller segments reveals that they are in fact more closely related: as shown in the tables above, their alleluia and jubilus segments are identical, and their verses are related by inversion with level 1 similarity. The similarity becomes more evident when examining the first phrase for the two chants. The larger segmentations are quite different, yet these phrases share level 2 similarity with a first order displacement—on par with the other chants, which display higher levels of similarity among larger segmentations. Even though the larger-scale analysis shows that this chant is very different, it is clear from this first verse segment that it still belongs in the same group.[20]

Generally speaking, the tables displayed in Figure 4.11 indicate only a moderate level of similarity between these chants in comparison with the *Dies sanctificatus* and the remaining eleven chants. In fact, *Dies sanctificatus* may not be the most typical representative of this melody. However, one can look at the high levels of similarity between certain chants' phrases and their constituent sections in order to create a hypothetical normative chant model to represent the chants in the list.

[20] The fact that Schlager and Hiley cite only nine chants that are the "same" as the *Dies sanctificatus* chant makes one wonder if they excluded the three chants that displayed a different overall prime. These may be different enough to seem like different melodies at first glance, but their similarities to the remaining ten chants within the smaller segmentations clearly illustrate their relationship to the chants in the list. Because of these similarities on these smaller scales, they should be included in the set along with the other chants that are classified as "the same," despite their surface-level differences.

In order to create this model, we must return to the surface level and compare the exact melodic content of each chant section.[21] We can take these melodic units (at the subphrase level) and use them to create a framework for a normative chant model. Figure 4.12 shows an abstract formal framework representing the 13 chants in question.[22] The chant is broken up into four sections of two to three units apiece. This framework provides a clear formal organization into which we can insert each chant under consideration.

Using this framework, one can compare corresponding subphrases within each section of the formal model. Most abstract sections illustrated in this model have 13 "real" subphrases associated with them, one for each of the 13 chants in the list. The task now is to decide which discrete subphrase is the most likely representative for each section. To decide the normative representative of all real subphrases within a section, one can look at the arithmetic mode of the section—the subphrase that appears most often in the list. For example, section 2.1 has 13 subphrases associated with it (the first verse subphrase of each of the 13 chants). Of these 13 subphrases, the subphrase ⟨10123454343121⟩ (seen first as the section *Video*, from *Alleluia Video caelos*, shown

[21] I am no longer using the MCRA or comparative process to create the normative chant model. The comparisons made in order to create the normative model instead are designed to look for identical chant subphrases at the surface of the music. Sections are listed in Figure 4.12 using c-seg notation only to facilitate comparison between the normative model and each distinct chant in the list.

[22] Some chants are missing sections, thus deviating from this abstract formal model, but the underlying structure still suggests that this model can be used to describe the content of all of these chants. The majority of the chants features the same phrase structure, and therefore will have matching sections within the framework. Those that do not possess this same phrase structure still have sections that match up with certain c-segs created by the sections of the larger chants. They are counted with the c-segs that they resemble most closely.

Figure 4.12. Formal framework for the 13 chants

<table>
<tr><td colspan="3">First Phrase</td></tr>
<tr><td colspan="2">1.1 (Alleluia)
Surface: ⟨01343421⟩
Prime ⟨021⟩</td><td>1.2 (Jubilus)
Surface: ⟨34540131323031231⟩
Prime ⟨2301⟩</td></tr>
<tr><td colspan="3">Second Phrase
(Beginning of the verse)</td></tr>
<tr><td colspan="2">2.1
Surface: ⟨10123454343121⟩
Prime ⟨1021⟩</td><td>2.2
Surface: ⟨1213453143201313120⟩
Prime ⟨120⟩</td></tr>
<tr><td colspan="3">Third Phrase
(Middle of the verse)</td></tr>
<tr><td colspan="2">3.1
Surface: ⟨21212120⟩
Prime ⟨10⟩</td><td>3.2
Surface: ⟨01310343231⟩
Prime ⟨021⟩</td></tr>
<tr><td colspan="3">Fourth Phrase
(End of the verse)</td></tr>
<tr><td>4.1
Surface:
⟨02343232010⟩
Prime ⟨010⟩</td><td>4.2
Surface:
⟨121345314320131312010⟩
Prime ⟨120⟩</td><td>4.3
Surface:
⟨34354543201231⟩
Prime ⟨2301⟩</td></tr>
</table>

in Figure 4.9a and 4.9d) appears eight times, and therefore represents the section. Taking the most common subphrase for each section in the formal framework, we can construct a musical representation of this hypothetical chant. Figure 4.13 shows the entire normative model chant, as it would appear in musical notation, featuring all of the "most common" subphrases within each section.[23]

[23] This is not meant in any way to represent a hypothetical source chant from which the 13 real chants emerged. It is simply meant to illustrate commonalities between each of the 13 chants in the list.

Figure 4.13a. Normative model chant

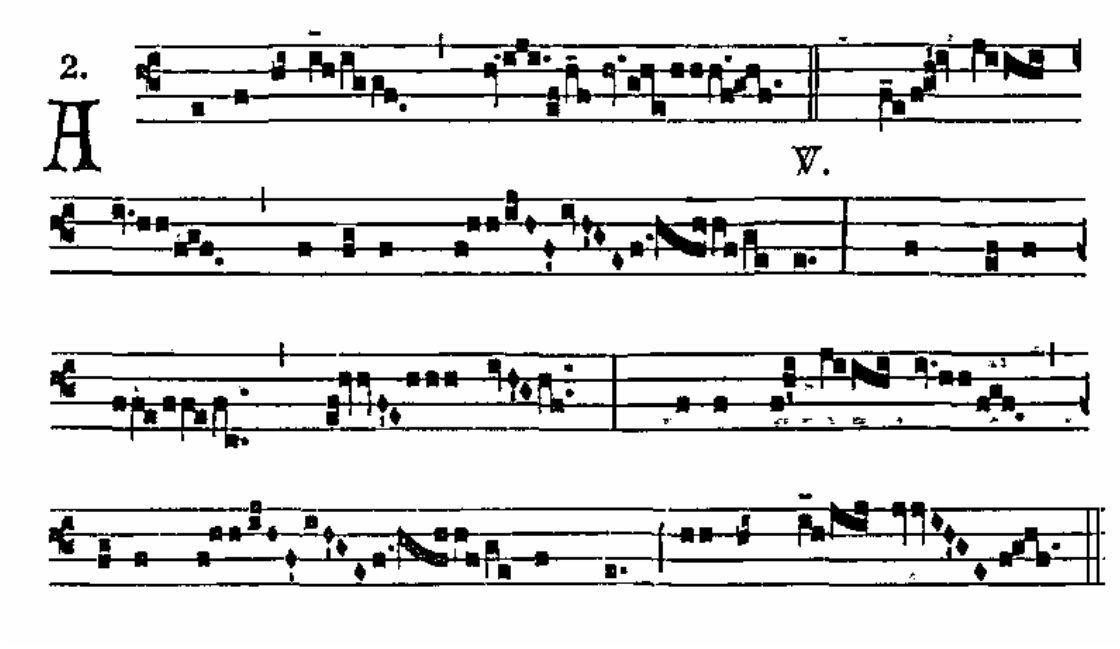


Figure 4.13b. Reduction of normative model chant

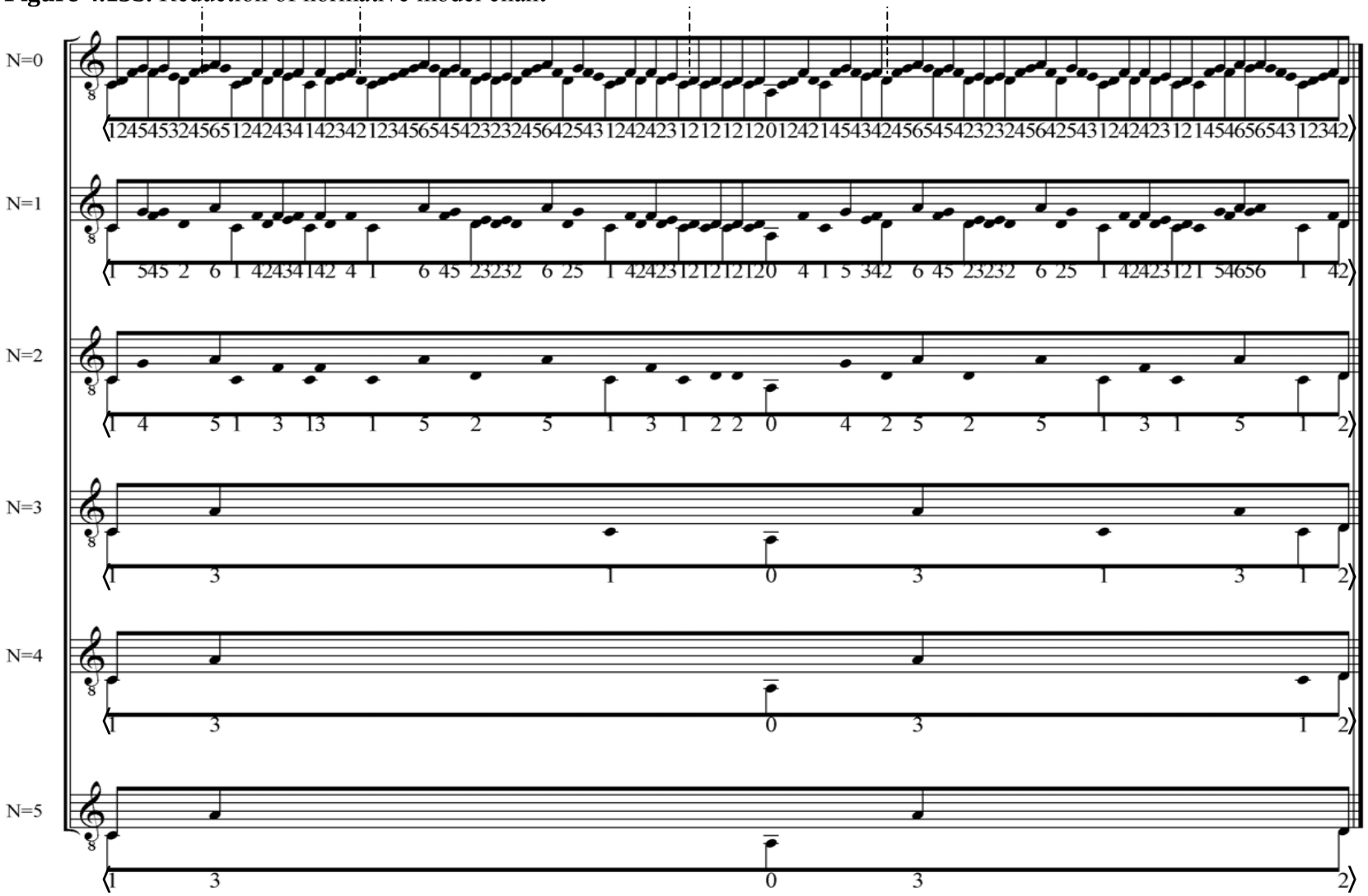


Figure 4.13c. Reduction of the verse from the normative model chant

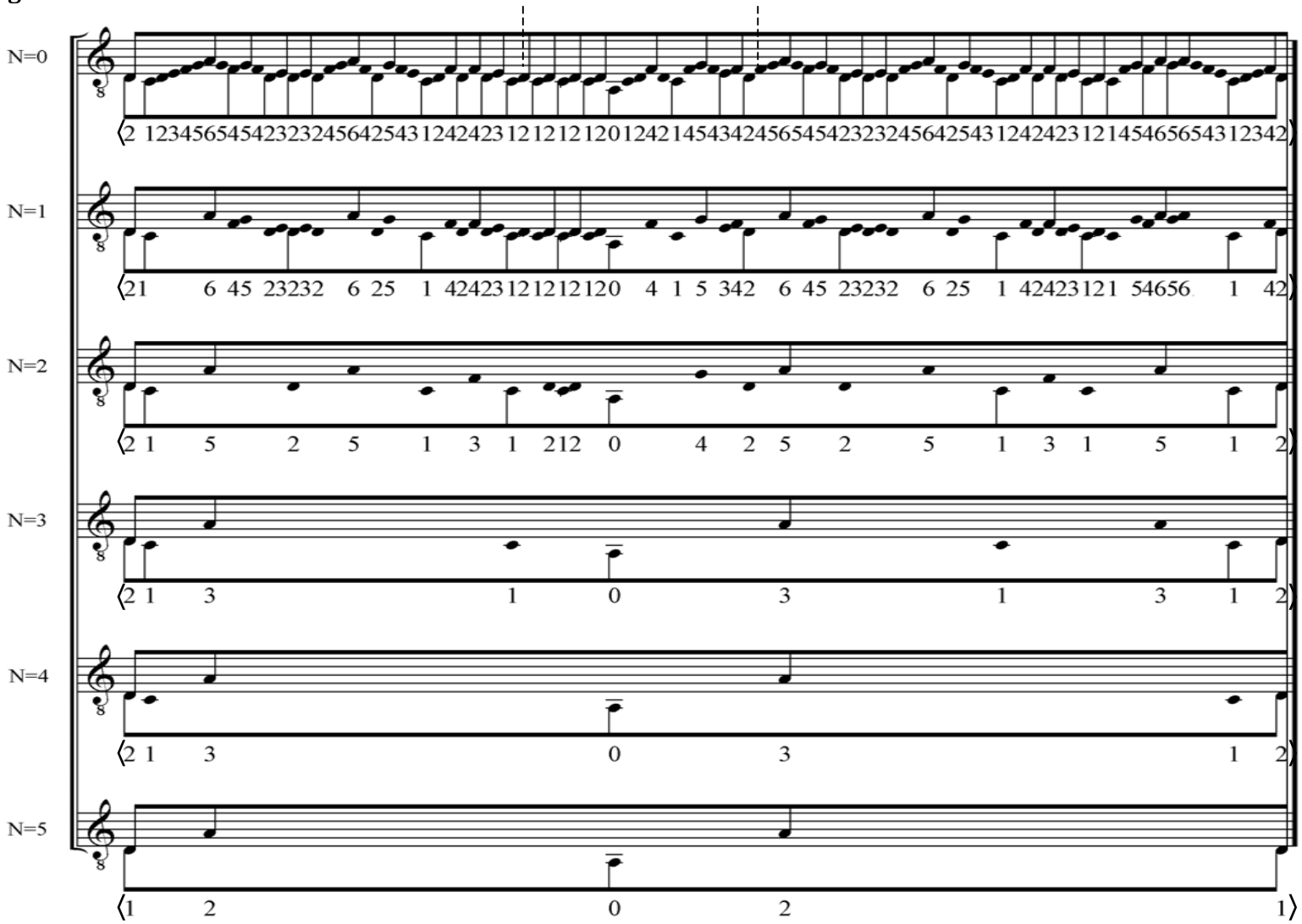


Figure 4.13d. Reduction of the first phrase from the normative model chant

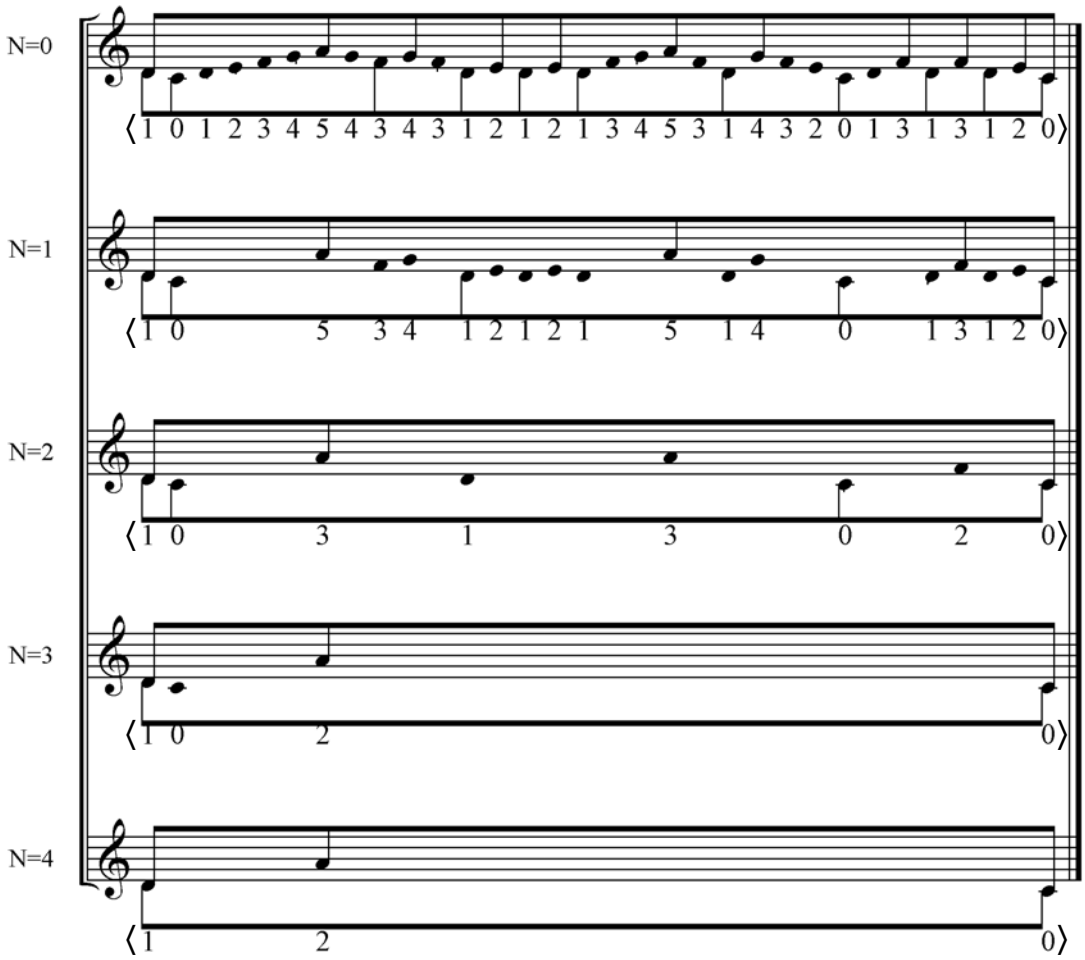


Figure 4.13e. Reduction of the second phrase from the normative model chant

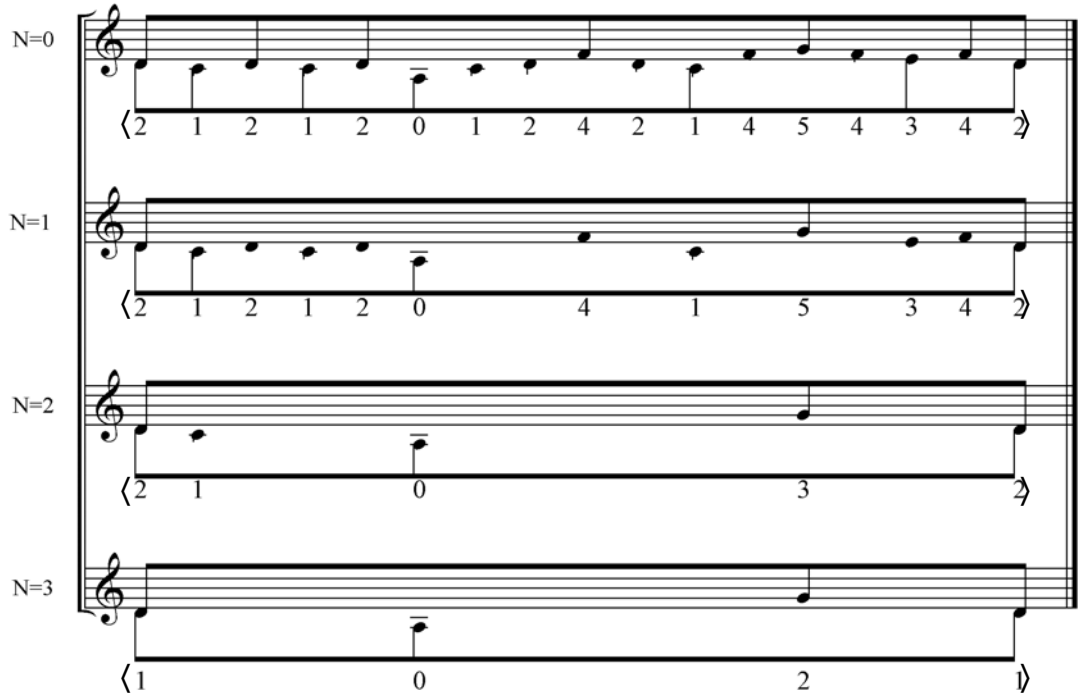
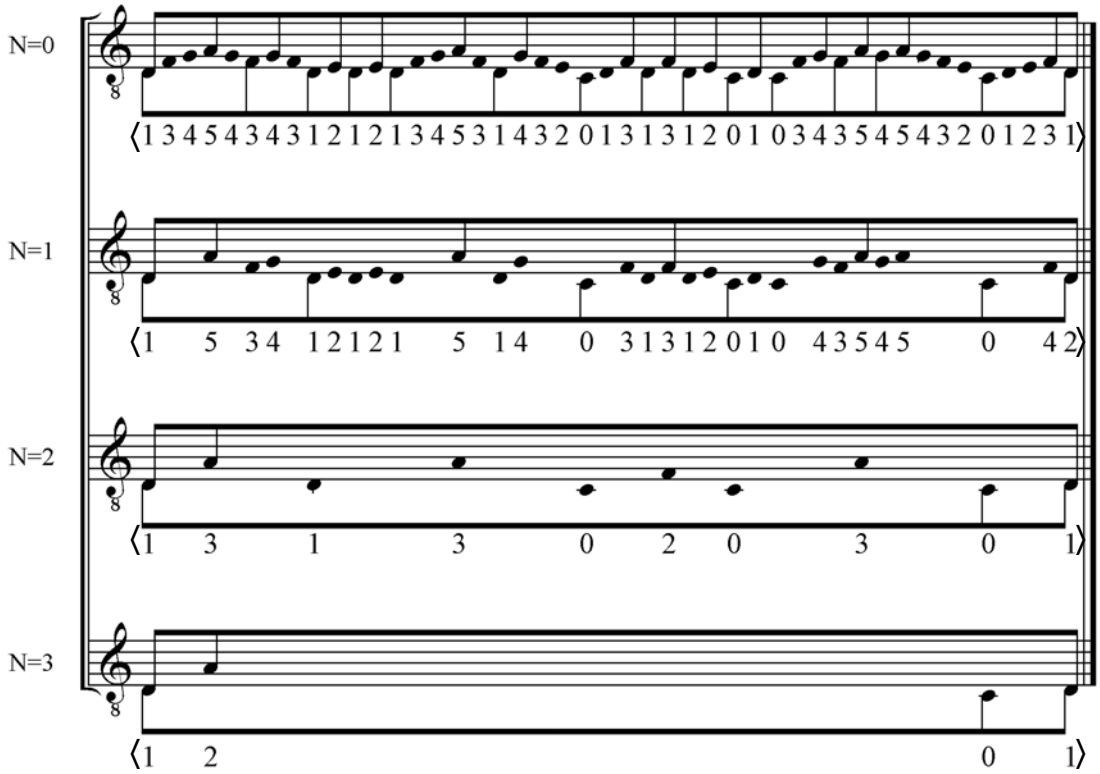


Figure 4.13f. Reduction of the third phrase from the normative model chant



Now that a normative chant has been formed, one can conduct comparative analyses again, using the normative model as the point of reference. The following tables in figure 4.14 display the similarity levels between the segments of the 13 chants when compared to this normative model.

The comparisons displayed in the tables above show a general increase in similarity between most of the chants and the model used in Figure 4.9, *Alleluia Dies sanctificatus*. However, of these increases in similarity, only one is identical to the normative chant in every case. This means that the remaining chants all have differences at levels closer to the surface that make them unique. If these chants are indeed called identical, what is it about these chants that is causing them to differ from the normative chant model? Why is it that they do not all possess the precise identity to which scholars such as Apel and Hiley have alluded?

Certainly the absence of entire verse-segments in the shorter alleluias of the list (chants *e, f, g,* and *j*) will contribute to the high degree of difference for these chants, and this points us in the direction of our answer. These shorter chants have fewer words—indeed, they lack an entire phrase of text—and therefore would not have as much of a need for extra melodic content. To retain the exact melodic content of the longer alleluias would create melismas that are both unnecessary and would break up the text in such a way that would detract from its meaning.

Lack of text accounts for the large differences in some chants, but what accounts for the surface and sub-surface level differences between the chants that possess a greater similarity to the normative chant model? To answer this question, let us return to the two

Figure 4.14a. Similarity comparison of entire chant c-segs with the normative chant model

	Chant name	Prime	Depth level	Similarity level	Similarity index
A	*Alleluia Dies sanctificatus*	⟨13032⟩	N=5	3	0.43
B	*Alleluia Video caelos*	⟨13032⟩	N=5	4	0.57
C	*Alleluia Hic est discipulus*	⟨13032⟩	N=5	5	0.71
D	*Alleluia Vidimus stellam*	⟨13032⟩	N=5	7	1.0
E	*Alleluia Redemptionem*	⟨021⟩	N=4	0	0
F	*Alleluia Tu es petrus*	⟨13032⟩	N=4	1	0.14
G	*Alleluia Hic est sacerdos*	⟨021⟩	N=4	0	0
H	*Alleluia Sancti tui domine*	⟨13032⟩	N=5	5	0.71
I	*Alleluia Magnus sanctus*	⟨13032⟩	N=5	3	0.43
J	*Alleluia Nunc com eo*	⟨021⟩	N=4	0	0
K	*Alleluia Inveni David*	⟨13032⟩	N=5	4	0.57
L	*Alleluia Tu puer propheta*	⟨13032⟩	N=5	4	0.57
M	*Alleluia Domine diligo*	⟨13032⟩	N=5	3	0.43

Figure 4.14b. Similarity comparison of verse c-segs with the normative chant model

	Chant name	Prime	Depth level	Similarity level	Similarity index
A	*Alleluia Dies sanctificatus*	⟨12021⟩	N=5	3	0.43
B	*Alleluia Video caelos*	⟨12021⟩	N=5	5	0.71
C	*Alleluia Hic est discipulus*	⟨12021⟩	N=5	5	0.71
D	*Alleluia Vidimus stellam*	⟨12021⟩	N=5	7	1.0
E	*Alleluia Redemptionem*	⟨10201⟩	N=4	1 (inversion)	0.14 (inversion)
F	*Alleluia Tu es petrus*	⟨23031⟩	N=4	0	0
G	*Alleluia Hic est sacerdos*	⟨10201⟩	N=3	1 (inversion)	0.14 (inversion)
H	*Alleluia Sancti tui domine*	⟨12021⟩	N=5	5	0.71
I	*Alleluia Magnus sanctus*	⟨12021⟩	N=5	4	0.57
J	*Alleluia Nunc com eo*	⟨2301⟩	N=3	0	0
K	*Alleluia Inveni David*	⟨12021⟩	N=5	5	0.71
L	*Alleluia Tu puer propheta*	⟨12021⟩	N=5	4	0.57
M	*Alleluia Domine diligo*	⟨12021⟩	N=5	3	0.43

Figure 4.14c. Similarity comparison of first phrase c-segs with the normative chant model

	Chant name	First Phrase	Prime	Depth level	Similarity level	Displacement	Similarity Index
A	*Alleluia Dies sanctificatus*	*Dies sanctificatus illuxit nobis*	⟨120⟩	N=5	2	1	0.4
B	*Alleluia Video caelos*	*Video caelos apertos*	⟨120⟩	N=4	6		1.0
C	*Alleluia Hic est discipulus*	*Hic est discipulus ille:*	⟨120⟩	N=4	6		1.0
D	*Alleluia Vidimus stellam*	*Vidimus Stellam ejus*	⟨120⟩	N=4	6		1.0
E	*Alleluia Redemptionem*	*Redemptionem misit dominus:*	⟨1021⟩	N=3	0		No sim
F	*Alleluia Tu es petrus*	*Tu es petrus et super hanc petram*	⟨120⟩	N=3	1		0.17
G	*Alleluia Hic est sacerdos*	*Hic est sacerdos*	⟨120⟩	N=4	4		0.67
H	*Alleluia Sancti tui domine*	*Sancti tui domine:*	⟨120⟩	N=4	4		0.67
I	*Alleluia Magnus sanctus*	*Magnus sanctus paulus, vas electionis*	⟨120⟩	N=4	3		0.5
J	*Alleluia Nunc com eo*	*Nunc cum eo*	⟨120⟩	N=2	1		0.17
K	*Alleluia Inveni David*	*Inveni David*	⟨120⟩	N=4	4		0.67
L	*Alleluia Tu puer propheta*	*Tu, puer propheta*	⟨120⟩	N=4	4		0.67
M	*Alleluia Domine diligo*	*Domini diligo*	⟨120⟩	N=4	4		0.67

Figure 4.14d. Similarity comparison of second phrase chant c-segs with the normative chant model[24]

	Chant name	Second Phrase	Prime	Depth level	Similarity level	Similarity Index
A	*Alleluia Dies sanctificatus*	*Venite gentes et adorate dominum*	⟨1021⟩	N=3	3	0.6
B	*Alleluia Video caelos*	*Et jesumstamtem*	⟨1021⟩	N=3	5	1.0
C	*Alleluia Hic est discipulus*	*Qui testimonium perhibet de his:*	⟨1021⟩	N=3	3	0.6
D	*Alleluia Vidimus stellam*	*In oriente*	⟨1021⟩	N=3	5	1.0
E	*Alleluia Redemptionem*	*In populosuo:*	⟨1021⟩	N=2	1	0.2
F	*Alleluia Tu es petrus*	*Aedificabo*	⟨1021⟩	N=3	5	1.0
G	*Alleluia Hic est sacerdos*	--	--	--	--	--
H	*Alleluia Sancti tui domine*	*Benedicent te:*	⟨1021⟩	N=3	3	0.6
I	*Alleluia Magnus sanctus*	*Verdigne est glorificandus:*	⟨1021⟩	N=3	3	0.6
J	*Alleluia Nunc com eo*	*Regnas:*	⟨120⟩	N=3	0	0
K	*Alleluia Inveni David*	*Servum meum:*	⟨1021⟩	N=3	3	0.6
L	*Alleluia Tu puer propheta*	*Altisimi vocaberis*	⟨1021⟩	N=3	3	0.6
M	*Alleluia Domine diligo*	*Habitaculum domus tuae*	⟨1021⟩	N=3	3	0.6

[24] Once again, three of the chants are shortened and therefore do not contain this middle verse segment.

Figure 4.14e. Similarity comparison of third phrase chant c-segs with the chant *Alleluia Dies sanctificatus*[25]

	Chant name	Third Phrase	Prime	Depth level	Similarity level	Similarity Index
A	*Alleluia Dies sanctificatus*	*Quia hodie descendit lux magna super terram*	⟨1201⟩	N=3	3	0.6
B	*Alleluia Video caelos*	*A dextris virtutis dei*	⟨1201⟩	N=3	3	0.6
C	*Alleluia Hic est discipulus*	*Et scimus quia verum est testimonium ejus*	⟨1201⟩	N=3	5	1.0
D	*Alleluia Vidimus stellam*	*Et venimus cum mune vibus adorare dominum*	⟨1201⟩	N=3	3	0.6
E	*Alleluia Redemptionem*	--	⟨2301⟩	--	0	0
F	*Alleluia Tu es petrus*	*Ecclesiam meam*	⟨1201⟩	N=2	1	0.2
G	*Alleluia Hic est sacerdos*	*Quem coronavit dominus*	⟨2301⟩	N=2	0	0
H	*Alleluia Sancti tui domine*	*Gloriam regni tui dicent*	⟨1201⟩	N=3	3	0.6
I	*Alleluia Magnus sanctus*	*Qui et meruit thronum duodecimum possidere*	⟨1201⟩	N=3	5	1.0
J	*Alleluia Nunc com eo*	*In aeternum*	⟨2301⟩	N=2	0	0
K	*Alleluia Inveni David*	*Oleo sancto meo unxieum*	⟨1201⟩	N=3	3	0.6
L	*Alleluia Tu puer propheta*	*Praeibis ante Dominum parare vias ejus*	⟨1201⟩	N=3	5	1.0
M	*Alleluia Domine diligo*	*Et locum tabernaculi gloriae tuae*	⟨1201⟩	N=3	3	0.6

[25] One of the chants is shortened, and does not have a unique third verse segment. Instead, the *Alleluia Redemptionem* repeats the jubilus, which is not out of the ordinary for an alleluia. In fact, the closing sections of the third verse segments within the longer chants also feature a ⟨2301⟩ prime, just like the jubilus (and the segment listed in the table for *Alleluia Redemptionem.*

chants discussed earlier: *Alleluia Dies sanctificatus* (hereafter known as chant *a*) and *Alleluia Video caelos* (hereafter known as chant *b*). The reductions of the c-segs for the first verse phrase are shown in Figure 4.15. The comparative process shows a level 1 similarity with first order displacement between the two. We see that the primes are identical, and that the sub-cseg immediately shallower than prime is also identical. However, the two differ at sub-cseg$_3$ and sub-cseg$_2$: chant *a* has a sub-cseg$_3$ of ⟨1041214030⟩, while chant *b* has a sub-cseg$_2$ of ⟨10313020⟩. The sole distinction between these two sub-csegs is the single intervening maximum—the 2 in the middle of the chant *a* sub-cseg. Further difference is seen in shallower levels of the two chants. The sub-cseg$_1$ of chant *a*, ⟨105341212121215140313120⟩, differs from the sub-cseg$_1$ of chant *b*, ⟨10534121215140313120⟩ in the number of repetitions of the "1-2" pair in the middle of each. The quantity of internal repetition is the point of difference between these two c-segs.

We must then return to the surface-level c-segs for both chants, and examine why chant *a* contains more "1-2" repetitions than chant *b*. The reason may lie in the text-music relationship of this phrase. In chant *b*, the phrase is *Video caelos apertos*, which has only three words and eight syllables. Chant *a*, on the other hand, has the phrase *Dies sanctificatus illúxit nóbis*, containing four words and twelve syllables. The syllabic addition of the word *sanctificatus* in the first half of the phrase of chant *a*, accounts for the extra "1-2" repetitions in the middle of the surface-level c-segs. Here, John Stevens

Figure 4.15a Reduction of the phrase *Dies sanctificatus illuxit nobis* from *Alleluia Dies sanctificatus*

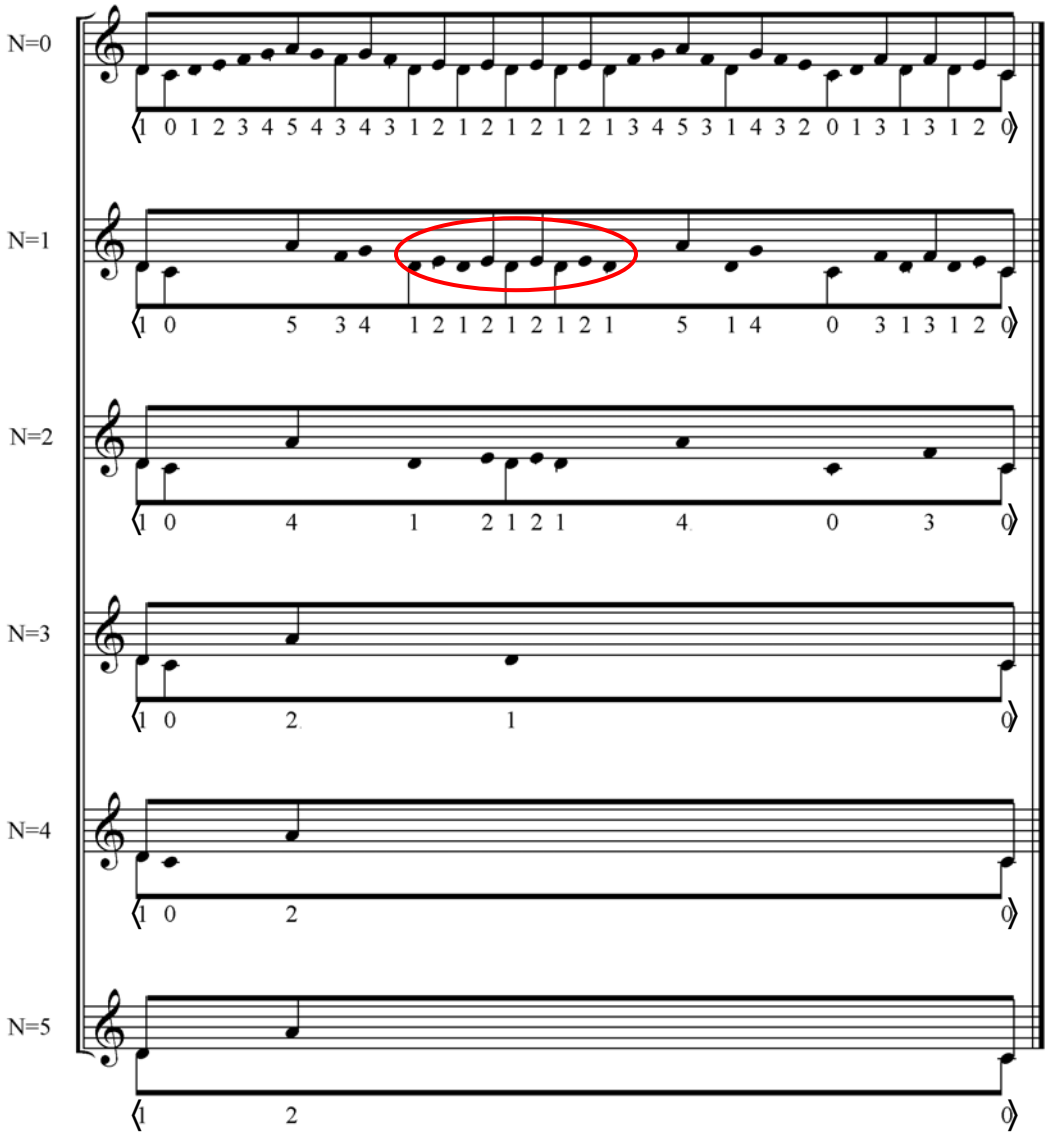
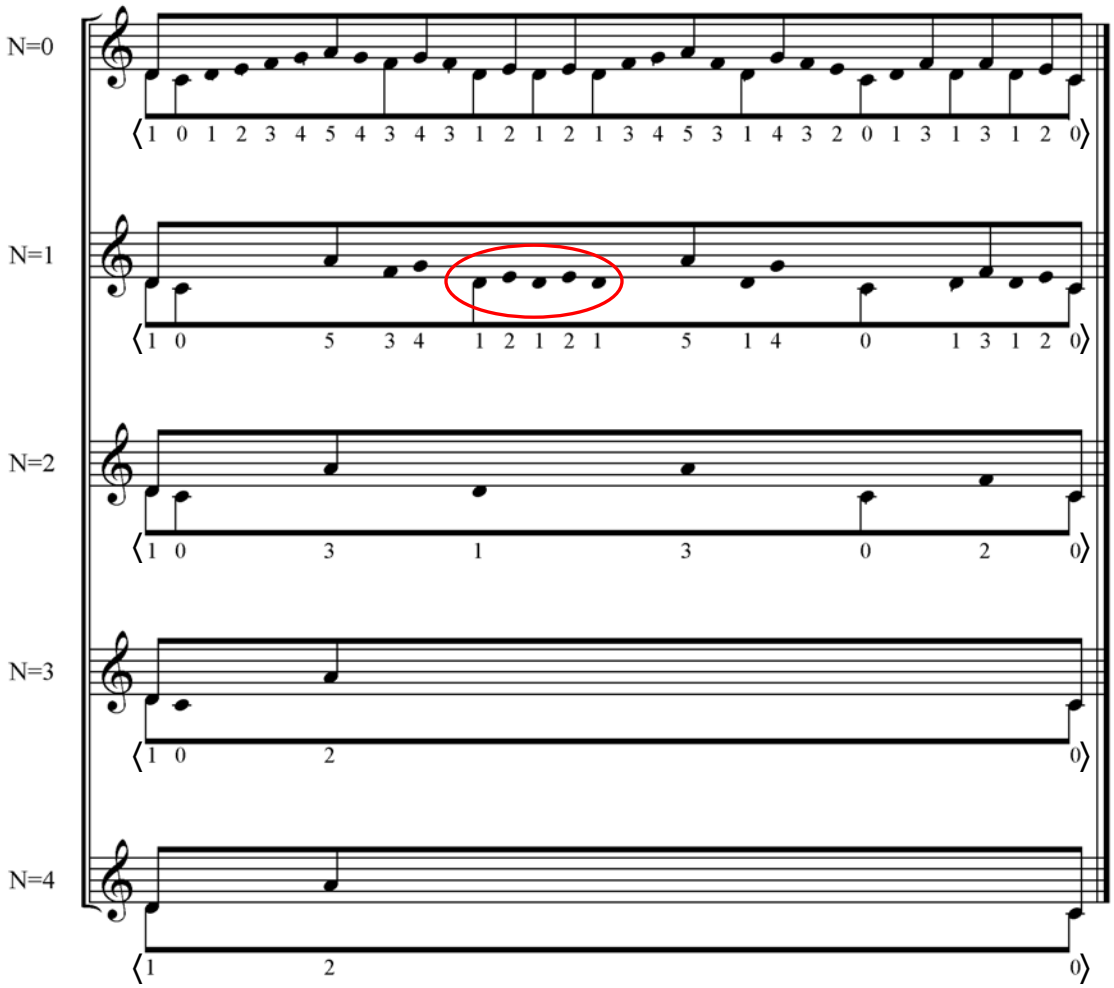
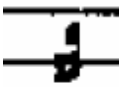


Figure 4.15b. Reduction of the phrase *Video caelos apertos* from *Alleluia Video caelos apertos*



suggests that accents in the text of the phrase are represented in the music by two possible means: "shortening or lengthening (accents of duration) and heightening (accents of pitch, usually called tonic accents)" (1986, 280). In the *sanctificatus* section, the word *sanctificatus* is begun with a liquescent podatus  , accounting for the first 1-2

segment, and the accent on the penultimate syllable creates a tonic accent in the music, and accounts for the second 1-2 podatus as Stevens suggests.

Another example of this accent relationship occurs when comparing c-segs of the first verse segments of chant *b* and chant *g (Alleluia Hic est sacerdos)*. These two chants share an even closer relationship than chants *a* and *b*: level 4 similarity. The primes occur on the same depth level, and both share a common sub-$cseg_3$ and sub-$cseg_2$. The difference is once again made apparent at the sub-$cseg_1$ level for both chants. The sub-$cseg_1$ for chant *b*, ⟨10534121215140313120⟩, differs from the sub-$cseg_1$ for chant *g*, ⟨105341215140313120⟩ again in the number of "1-2" repetitions there are. In chant *b* there are two, while in chant *g* there is only one—not enough even to qualify it as the same type of repeating pattern. The difference itself then is similar to the difference between chants *a* and *b*. However, on the surface-level of the music, this difference occurs not in the first half of the phrase, as did the comparison between chants *a* and *b*, but rather at the beginning of the second half of the phrase. The second half of the phrase in chant *b* has two words (*caelos apertos*) while the second half of the phrase in chant *g* has only one word: *sacerdos*. This is evident in the fact that the first syllable tonic accent on *cae* is no longer present. Instead, the accent on the second syllable of *sacerdos* is used as the starting point for a melisma containing the rest of the segment.[26] This is an exemplary illustration of Stevens's remark about the tonic accent. Here we see the presence of two accents in the *caelos apertos*, and only one in sace*rdos*: these accents (or lack thereof) have the power to alter the contour of the c-segs in question in these areas. The text has the power to modify, or in some cases even override the normative contour

[26] This is the same melisma that occurs over *apertos* in chant *b*.

model, accounting for the differences between it and each chant in the list, as well as between the chants themselves.

The relationships discussed above would not be as readily apparent, or as accurately explained without the MCRA and hierarchical method of comparison. Comparing intermediary sub-csegs between these 13 chants reveals differences in the way the primes that govern their basic structures are composed out (to borrow a term from Schenker). Using the hierarchical comparison method, we can arrive at a more precise quantitative measurement of difference within the 13 chants. Such measurements are useful tools for the determination of difference between interchangeable units within a paradigmatic style of analysis, and for exploring the various reasons for these differences. In the case of these 13 mode-2 Alleluias with the same alleluia and jubilus, the comparison reveals surface and sub-surface level differences that arise from differences in both the length of the text, and the textual accents that occur within the Latin language. The hierarchy in this case illustrates the importance of these textual accents in the role of creating intricate contour structures across levels not commonly found in simpler chants.

CHAPTER 5

CONCLUSION

In the previous chapters of this thesis, I have presented refinements to the theory of melodic contour and the reductive approach of the MCRA. The MCRA provides a hierarchy of structural levels as a result of the recursive pruning procedure contained therein. This thesis has extended the analytical capabilities of the MCRA by providing a methodology for examining the hierarchical levels between the prime and the surface levels. It uses the levels within the hierarchy to refine the comparison between two c-segs, extending the capabilities of the MCRA to allow for such comparisons to occur.

A simple glance at the hierarchical levels within related c-segs suggests the need for adjustment of the comparison process when using the MCRA. C-segs may not only have identical primes, but they may or may not also feature identical sub-csegs on levels shallower than prime. As such, a method is needed to quantify the degree to which two c-segs with the same prime are similar. In Chapter 2 of this thesis, I have introduced a concise methodology that enhances the comparison process of c-segs using the MCRA. This comparison method provides a structured step-by-step approach for comparing sub-csegs within two given c-segs that share the same complexity (i.e., they reduce to prime at the same depth level). The direct comparison evaluates sub-csegs at equivalent depth levels in order to determine whether they are identical, and arrives at an indicator of similarity by counting the number of levels that are found to be identical. I have taken this similarity level and created an index value representing the degree to which the two c-segs in question are similar.

Although this methodology allows for the comparison of sub-csegs in c-segs with the same prime and complexity, it requires the depth levels of the sub-csegs to be equal; it therefore lacks the ability to make comparisons across depth levels. The comparison method developed in Chapter 3 fills this void by introducing the notion of displaced similarity.

In the displaced comparison, the prime's depth levels are not equal: one lies closer to the surface than the other. The comparison process is similar to that of the direct comparison, but with a few crucial modifications. Both compare sub-csegs by methodically proceeding from the deepest level (prime) toward the surface level. However, in the displaced comparison, the depth levels of these sub-csegs do not align with one another. Instead, one would compare, for instance, sub-cseg$_3$ of one c-seg with sub-cseg$_2$ of the other. As such, the shallower c-seg's prime and any other corresponding sub-csegs can be conceived as embedded within the more complex c-seg. Since the displacement of the primes obscures the perception of similarity for a pair of c-segs, a new variable is introduced (*d*) that calculates the difference in complexity between the two c-segs. This value allows one to make a comparison of intermediary sub-csegs regardless of depth level, while also reflecting that the two c-segs are less similar than two c-segs whose complexity values are identical.

The study presented in Chapter 4 illustrated the application of the two comparative methods. It examined 13 mode-2 Alleluias that previous scholars (Apel 1958; and Hiley 1993) have grouped together as highly similar. It implemented three different lengths of segmentation, and ran the MCRA on each resulting c-seg. In this analysis, the comparison method provided a precise quantitative measurement of

difference between these c-segs. The comparisons made at each length of segmentation highlighted surface and sub-surface level differences between specific chants, which resulted from changes in both the length of each chant, and the change in textual accent between different words in the Latin language. Without the MCRA and the comparative methodology I have devised, the relationships between chants would not be as quantifiable, and explanations of the differences exhibited by these chants would be less systematic.

Further Research

The hierarchical comparison method has supplemented the way c-segs are compared using the MCRA, and therefore contributed to our understanding of how contour is portrayed in music. However, there are still areas for further refinement. First of all, differences between the direct and displaced comparisons could be more concisely defined, and an eventual unification of these two processes would allow for a more elegant analytical model. In addition, just as the hierarchical comparison expanded the notion of comparing c-segs using the MCRA, the idea of the comparison process could be further expanded to compare sub-csegs that are judged not to be equivalent. For example, if two c-segs share a level-3 similarity with a similarity index of 3/5 (0.6), this would indicate that two pairs sub-csegs are judged to be different. However, the notion of sub-cseg equivalence as a determinant of similarity excludes the possibility of high levels of similarity between these sub-csegs. A further refinement of the comparative process could be to apply Marvin and Laprade's (1987) CSIM or ACMEMB toward

these sub-csegs in order to arrive at a percentage of similarity for that specific level.[27] One could then combine this percentage in some fashion with the similarity index in order to arrive at a more precise quantification of similarity.

Another possible avenue of further refinement involves the extension of the hierarchical comparison to c-segs whose primes are related by inversion, retrograde, and retrograde inversion. If two primes are related by one of these transformations, it stands to reason that one or more of their intermediary sub-csegs may also be related in the same fashion. Just as the hierarchical comparison method compares sub-csegs based on the criteria of equivalence, a similar hierarchical comparison method could be used to compare sub-csegs based on these transformations. Such a methodology could then be integrated with the existing hierarchical comparisons to lend further analytical capability to the comparison technique.

Finally, further research could include the integration of the hierarchical comparison method with other existing theories of musical contour. One might adapt the method so that it can be applied to the window algorithms of Bor's (2009) reductive technique. Since the hierarchical structures are created in different ways under Bor's method, the hierarchical comparison process would have to be modified in order to be of use in that analytical setting. Other contour theories that might benefit from the hierarchical comparison process include Schultz's (2009) methods for examining musical contour diachronically. Exploring how c-segs compare with each other as each pitch is

[27] In terms of examining c-segs for similarity, CSIM would be the more applicable choice. However, CSIM is only applicable toward c-segs of the same cardinality, and therefore would not be applicable in many cases. A subsequent refinement of CSIM would need to occur before it could be used in this fashion consistently.

added could provide further avenues of refinement for the hierarchical comparison process.

Future developments in any of these directions would prove valuable to the success of the comparative method I have devised, and to the overall usefulness of the MCRA in general. Certainly these tools can provide valuable insights into multiple styles of music across many musical and historical eras, and this would contribute much to the existing set of tools within the domain of contour theory. As I have shown throughout this thesis, the refinement of the comparison process makes resulting analyses stronger, and therefore enhances our understanding of melodic construction.

APPENDIX

THE THIRTEEN COMMON MODE 2 ALLELUIAS

Figure A.1. ***Alleluia Dies Sanctificatus***

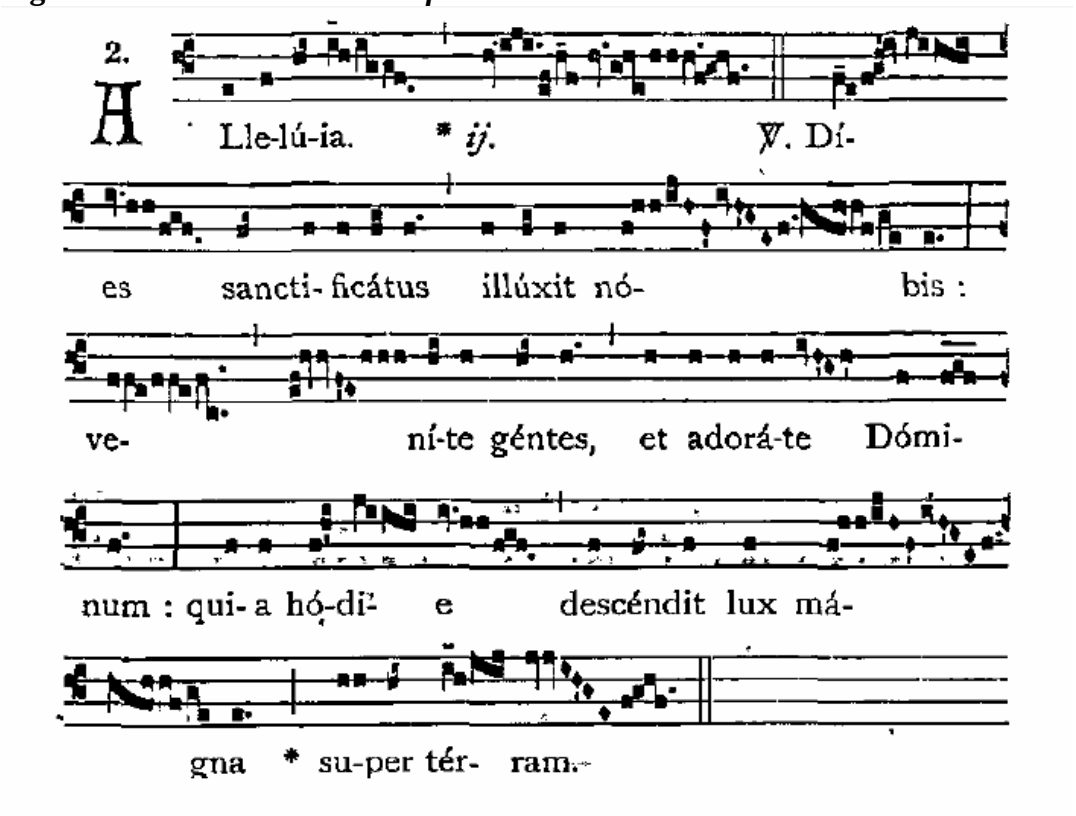


Figure A.2. *Alleluia Video Caelos*

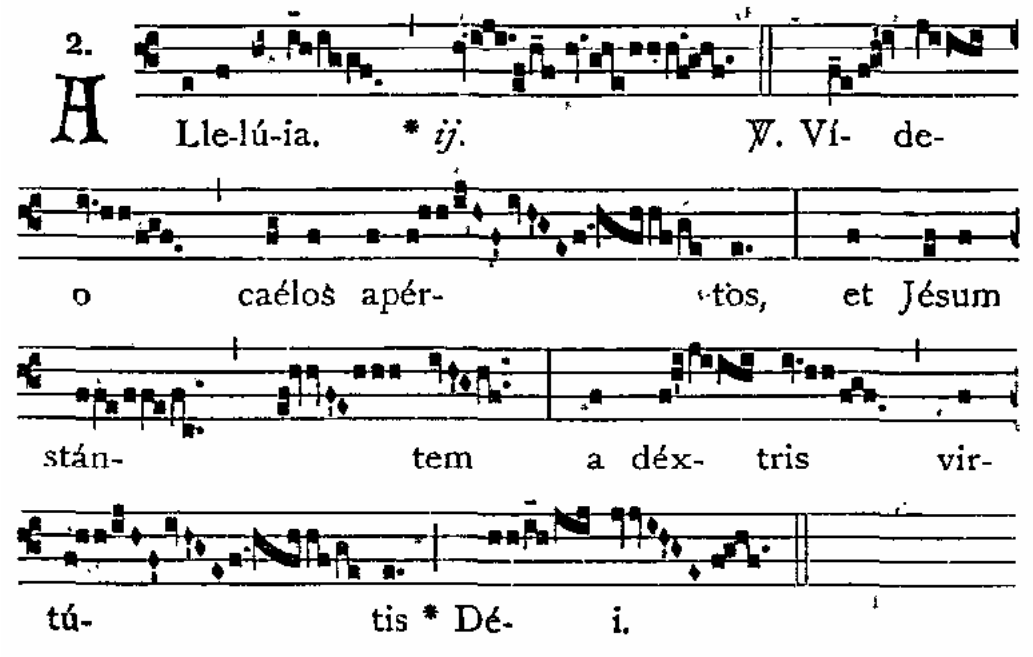


Figure A.3. *Alleluia Hic est discipulus*

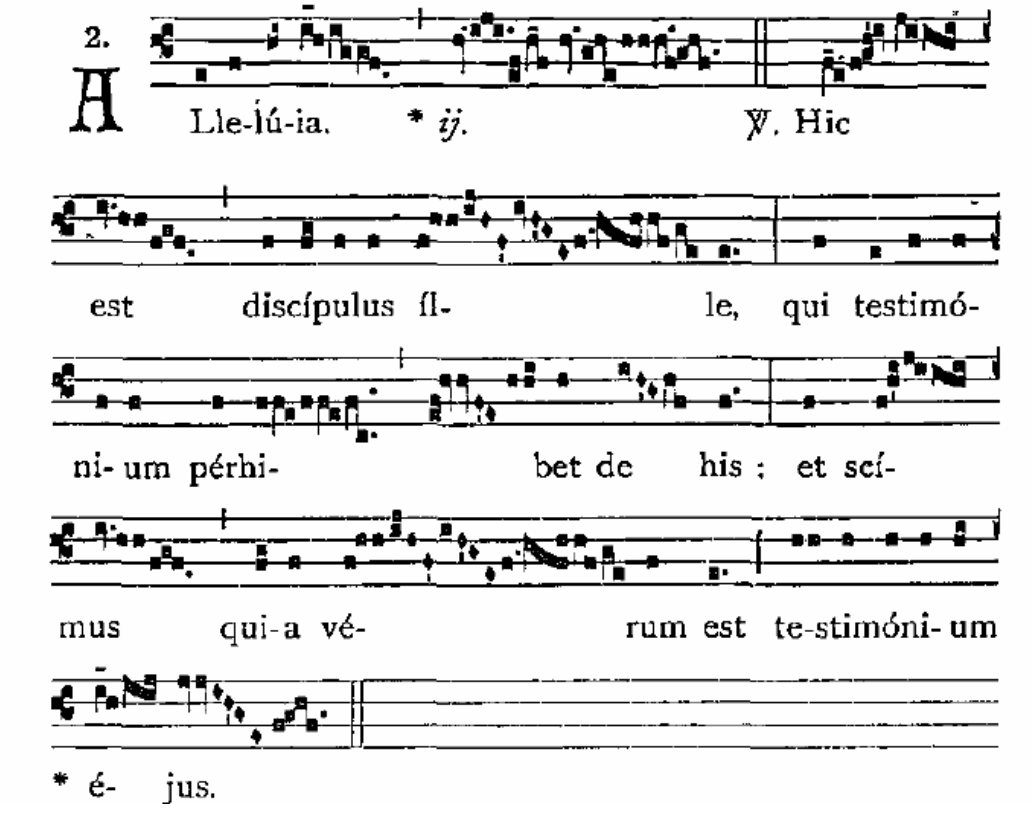


Figure A.4. *Alleluia Vidimus stellam*

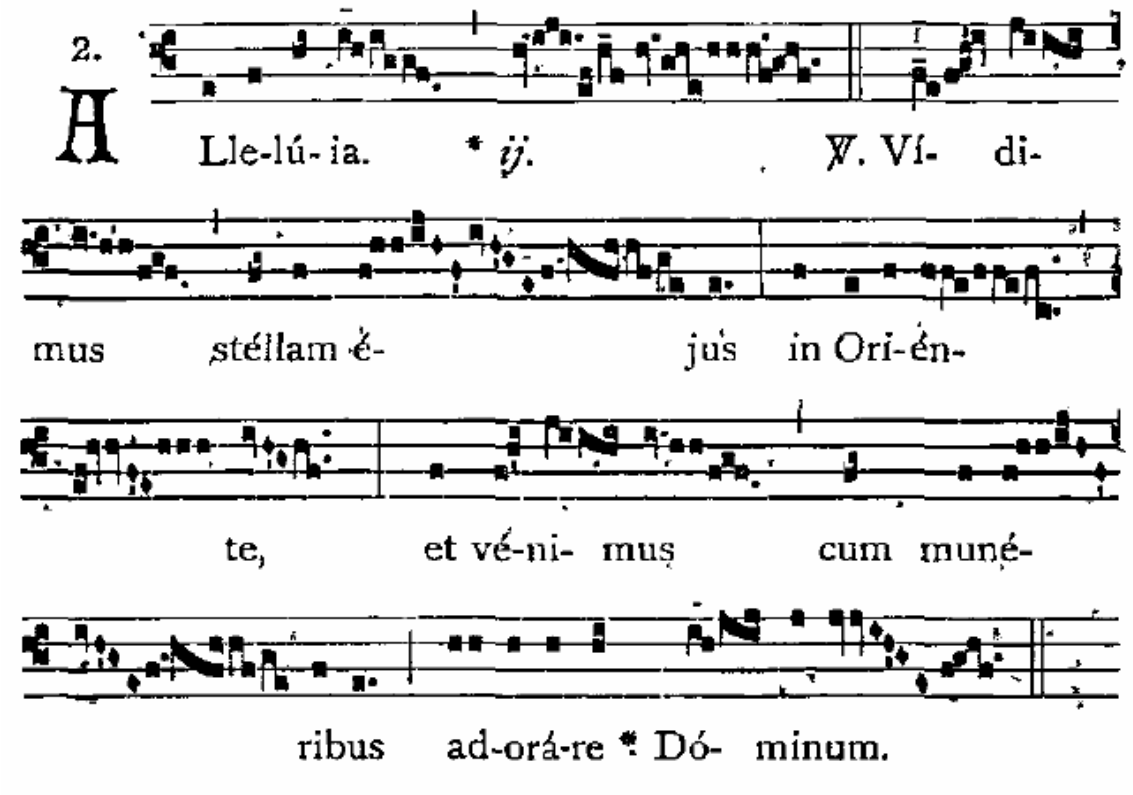


Figure A.5. *Alleluia Redemptionem*

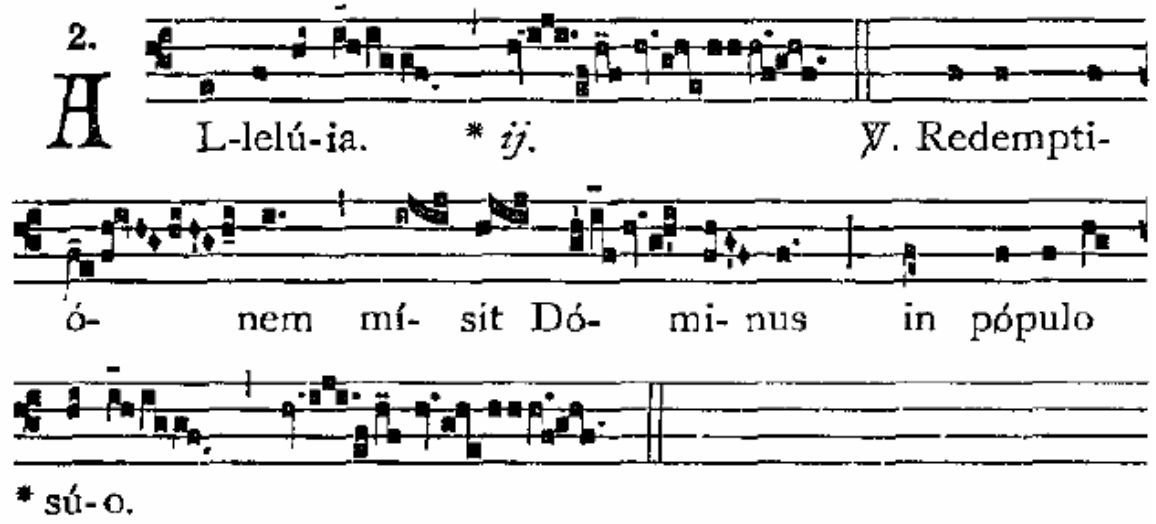


Figure A.6. *Alleluia Tu es Petrus*

2. A Lle-lú-ia. * ij. ℣. Tu es Pé-
trus, et super hanc pé- tram aedi-fi-
cá- bo Ecclé-si-am * mé- am.

Figure A.7. *Alleluia Hic est sacerdos*

2. A Lle-lú-ia. * ℣. Hic
est sacér- dos, quem co-roná-vit
* Dó- minus.

Figure A.8. *Alleluia Sancti tui*

2. Al-le-lú-ia. * *ij.* ℣. Sán-cti tú-
i, Dó- mi-ne, be-
nedí-cent te : gló-ri- am ré- gni
tú- i * dí- cent.

Figure A.9. *Alleluia Magnus sanctus*

2. Al-le-lú-ia. * *ij.* ℣. Mágnus sán-
ctus Páu- lus, vas e-lectí- ó-
nis, vere di- gne est glo-ri-fi- cándus,
qui et mé-ru- it thrónum du-odé-
cimum * pos-sidé- re.

Figure A.10. *Alleluia Nunc cum eo*

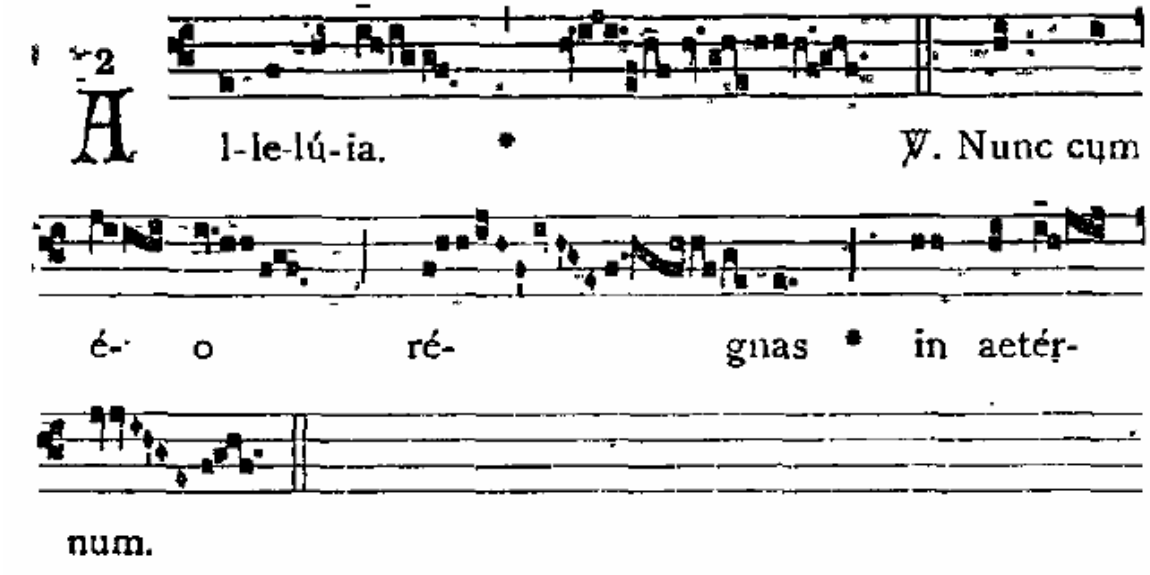


Figure A.11. *Alleluia Inveni David*

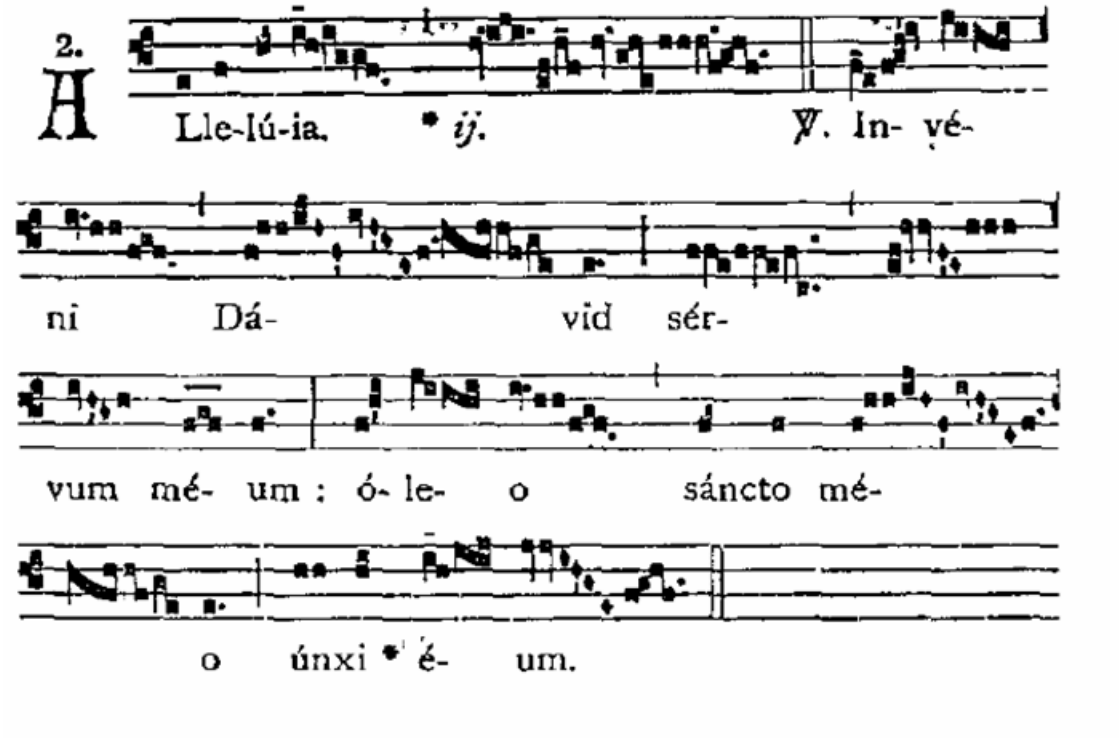


Figure A. 12. *Alleluia Tu puer Propheta*

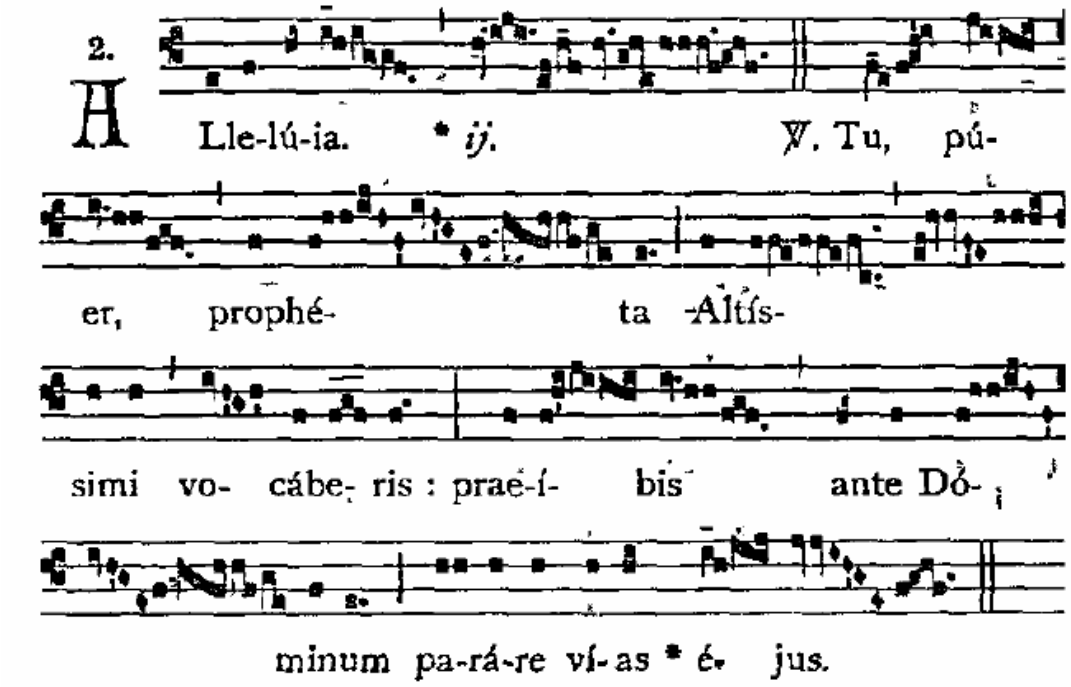
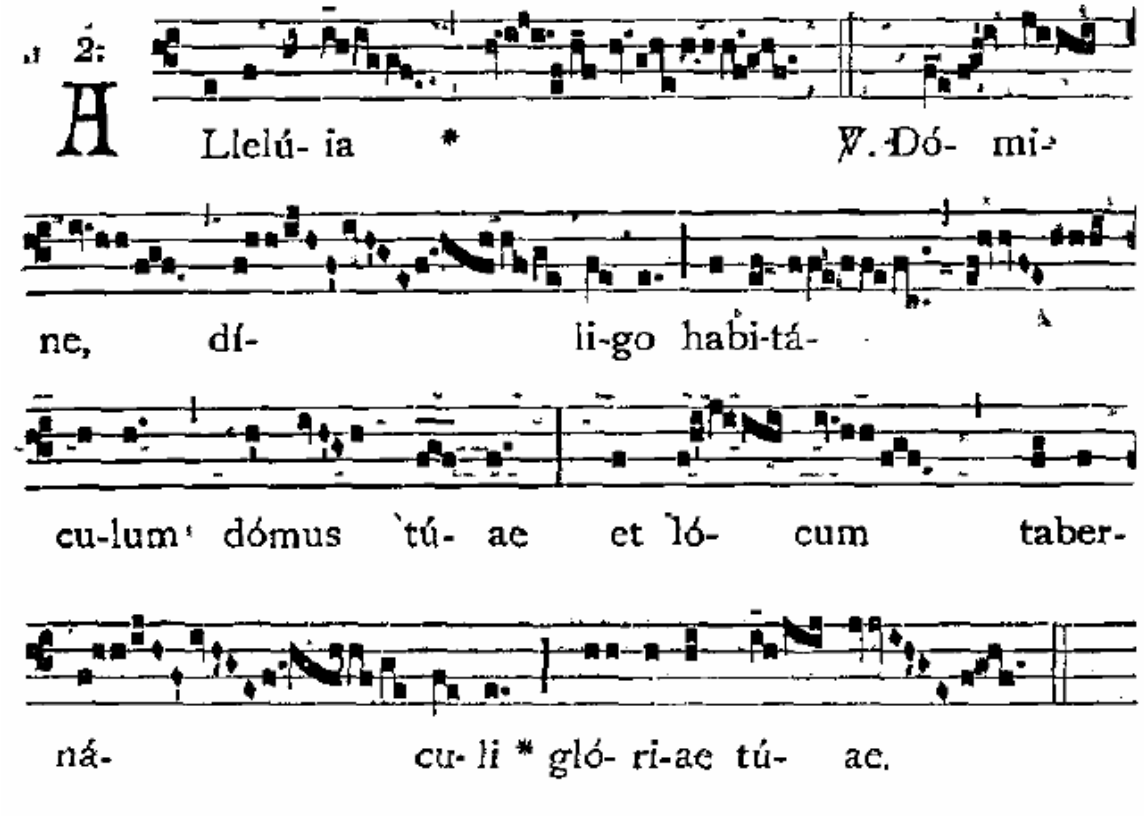


Figure A. 13. *Alleluia Domine diligo*



REFERENCES

Adams, Charles. 1976. "Melodic Contour Typology." Ethnomusicology 20.2: 179–215.

Apel, Willi. 1958. *Gregorian chant*. London: Burns & Oates.

Bor, Mustafa. 2009. "Contour Reduction Algorithms: A Theory of Pitch and Duration Hierarchies For Post-Tonal Music." Ph.D diss., University of British Columbia. 48-110.

Brahms, Johannes. 2008. *Sonate für Klavier und Violine G-Dur, Opus 78.* ed. Hans Otto Hiekel, Wolfgang Sandberger, Hans-Martin Theopold, and Karl Röhrig. München: G. Henle.

Catholic Church. 1961. *The Liber Usualis: with Introduction and Rubrics in English*. Tournai: Desclee Co.

———. 1961. *Graduale sacrosanctae Romanae Ecclesiae: de tempore et de sanctis: SS. D. N. Pii X. Pontificis Maximi*. Tournai: Desclee Co.

———. 1979. *Graduale triplex: seu Graduale Romanum Pauli PP. VI cura recognitum & rhythmicis signis a Solesmensibus monachis ornatum.* Solesmis: Abbaye Saint-Pierre de Solesmes.

Friedmann, Michael L. 1985. "A Methodology for the Discussion of Contour: Its Application to Schoenberg's 'Music'." Journal of Music Theory 29.2: 223-48.

———. 1987. "A Response: My Contour, Their Contour." *Journal of Music Theory* 31.2: 268–74.

Hiley, David. 1993. *Western plainchant: a handbook*. Oxford: Clarendon Press.

Hucbald, Guido, Johannes, Warren Babb, and Claude V. Palisca. 1978. *Hucbald, Guido, and John on music: three medieval treatises*. New Haven: Yale University Press.

Marvin, Elizabeth West, and Paul A. Laprade. 1987. "Relating Musical Contours: Extensions of a Theory for Contour." Journal of Music Theory 31.2: 225-67.

Marvin, Elizabeth West. 1995. "A Generalization of Contour Theory to Diverse Musical Spaces: Analytical Applications to the Music of Dallapiccola and Stockhausen." *Concert Music, Rock, and Jazz Since 1945*, ed. Elizabeth West Marvin and Richard Herman. Rochester: Rochester University Press. 135–171.

Morris, Robert. 1987. *Composition with Pitch-classes: a Theory of Compositional Design.* New Haven: Yale University Press.

———. 1993. “New Directions in the Theory and Analysis of Musical Contour.” Music Theory Spectrum 15.2: 205-28.

Robertson, Alec. 1970. *The Interpretation of Plainchant*. Westport CT: Greenwood Press.

Schlager, Karlheinz. 1968. *Alleluia-Melodien I bis 1100*. Kassel: Bärenreiter.

Schoenberg, Arnold. 1967. *Fundamentals of Musical Composition*. Edited by Gerald Strang. New York: St. Martin’s Press.

Seeger, Charles. 1960. “On the Moods of a Music-Logic.” Journal of the American Musicological Society 13.1: 224–261.

Schultz, Rob. 2008. “Melodic Contour and Nonretrogradable Structure in the Birdsong of Olivier Messiaen.” Music Theory Spectrum 30.1: 87-139

———. 2009. “A Diachronic-Transformational Theory of Musical Contour Relations.” Ph.D. diss., University of Washington.

Stevens, John. 1986. *Words and Music in the Middle Ages*. Cambridge: Cambridge University Press.

Toch, Ernst. 1948. *The Shaping Forces in Music*. New York: Criterion Music Corp.

Wagner, Peter. 1986. *Introduction to the Gregorian Melodies: A Handbook of Plainsong*. New York: Da Capo Press.

Wagner, Peter. 1970. *Einführung in Die Gregorianischen Melodien* 3rd ed. *Gregorianische Formenlehre*. Vol. III. Weisbaden: Breitkopf & Härtel.