5-7-2011

Assessing the Impact of Computer Programming in Understanding Limits and Derivatives in a Secondary Mathematics Classroom

Christopher H. de Castro
Georgia State University

ACCEPTANCE

This dissertation, ASSESSING THE IMPACT OF COMPUTER PROGRAMMING IN UNDERSTANDING LIMITS AND DERIVATIVES IN A SECONDARY MATHEMATICS CLASSROOM, by CHRISTOPHER HAROLD DE CASTRO, was prepared under the direction of the candidate's Dissertation Advisory Committee. It is accepted by the committee members in partial fulfillment of the requirements for the degree Doctor of Philosophy in the College of Education, Georgia State University.

The Dissertation Advisory Committee and the student's Department Chair, as representatives of the faculty, certify that this dissertation has met all standards of excellence and scholarship as determined by the faculty. The Dean of the College of Education concurs.

Christine D. Thomas, Ph.D.
Committee Chair

Laurie Brantley-Dias, Ph.D.
Committee Member

Mary B. Shoffner, Ph.D.
Committee Member

George Davis, Ph.D.
Committee Member

Date

Dana L. Fox, Ph.D.
Chair, Department of Middle-Secondary Education and Instructional Technology

R.W. Kamphaus, Ph.D.
Dean and Distinguished Research Professor
College of Education

AUTHOR'S STATEMENT

By presenting this dissertation as a partial fulfillment of the requirements for the advanced degree from Georgia State University, I agree that the library of Georgia State University shall make it available for inspection and circulation in accordance with the regulations governing materials of this type. I agree that permission to quote, to copy from, or to publish this dissertation may be granted by the professor under whose direction it was written, by the College of Education's director of graduate studies and research, and by me. Such quoting, copying, or publishing must be solely for scholarly purposes and will not involve financial gain. It is understood that any copying from or publication of this dissertation which involves potential financial gain will not be allowed without my written permission.

Christopher Harold de Castro

NOTICE TO BORROWERS

All dissertations deposited in the Georgia State University Library must be used in accordance with the stipulations prescribed by the author in the preceding statement. The author of this dissertation is:

Christopher Harold de Castro
6130 Yorkridge Drive
Alpharetta, Georgia 30005

The director of this dissertation is:

Dr. Christine D. Thomas
Department of Middle-Secondary and Instructional Technology
College of Education
Georgia State University
Atlanta, Georgia 30303-3083

VITA

Christopher Harold de Castro

ADDRESS: 6130 Yorkridge Drive
Alpharetta, Georgia 30005

EDUCATION:

Ph.D.	2011	Georgia State University Teaching and Learning – Mathematics Education
M.Ed.	2007	Georgia State University Mathematics Education
M.S.	1990	Georgia Institute of Technology Electrical Engineering
B.	1989	Georgia Institute of Technology Electrical Engineering
B.S.	1989	Georgia Institute of Technology Mathematics

PROFESSIONAL EXPERIENCE:

1998-Present	Teacher Duluth High School, Duluth, GA
1997-1998	Teacher Marietta High School, Marietta, GA

PROFESSIONAL SOCIETIES AND ORGANIZATIONS

1997-Present	Mathematical Association of America
1997-Present	Professional Association of Georgia Educators

PRESENTATIONS AND PUBLICATIONS:

Brantley-Dias, L., Kinuthia, W., Shoffner, M.B., de Castro, C., & Rigole, N. (2007). Developing Pedagogical Technology Integration Content Knowledge in Preservice Teachers: A Case Study Approach. *Journal of Computing in Teacher Education*, 23(4).

Jha, R., Muhammed, M., Yalamanchili, S., Schwan K., Ivan, D., & de Castro, C. (1996). Adaptive Resource Allocation for Embedded Applications. *Proceedings of the International Conference on High Performance Computing*

de Castro, C., & Yalamanchili, S., (1995) Partitioning Coarse-Grain Signal Flow Graphs for Heterogeneous DSP Architectures. *International Journal in Computer Simulation special issue on Simulation of Signal Processing Circuits and Systems*..

de Castro, C., & Yalamanchili, S. (1993). Partitioning Algorithms for a Class of Application-Specific Multiprocessor Architectures. *Proceedings of the Workshop on Heterogeneous Processing at the 1993 International Parallel Processing Symposium*.

de Castro, C., & Yalamanchili, S. (1992). Partitioning Signal Flow Graphs for Execution on Heterogeneous Signal Processing Architectures. *Proceedings of the Workshop on Heterogeneous Processing at the 1992 International Parallel Processing Symposium*.

ABSTRACT

ASSESSING THE IMPACT OF COMPUTER PROGRAMMING
IN UNDERSTANDING LIMITS AND DERIVATIVES
IN A SECONDARY MATHEMATICS
CLASSROOM
by
Christopher H. de Castro

This study explored the development of student's conceptual understandings of limit and derivative when utilizing specifically designed computational tools. Fourteen students from a secondary Advanced Placement Calculus AB course learned and explored the limit and derivative concepts from differential calculus using visualization tools in the Maple computer algebra system. Students worked in pairs utilizing the pair-programming model of collaboration. Four groups of student pairs from one intact class programmed their own computational tools and subsequently used them to explore the limit and derivative concepts. Four additional pairs of students from an additional intact class were provided with similar pre-constructed computational tools and asked to perform identical explorations.

A multiple embedded case design was utilized to explore ways students in the two classes, programming class, P, and non-programming class, N, constructed understandings focusing upon their interactions with each other and with the computational tools. The Action-Process-Object-Schema (APOS) conceptual model and Constructionist framework guided design and construction of the tools, outlined developmental goals and milestones, and provided interpretive context for analysis.

The results provided insights into the effective design and use of computational tools in fostering conceptual understanding. The study found learning programming was

challenging and overburdened students in class P in ways that misdirected students' attention away from the intended mathematical concept of limit. Students in class P tended to see the limit as an unreachable boundary whereas students in class N, using pre-constructed exploratory tools, tended to see the limit in its proper formal form. The study additionally found, however, that pre-constructed tools could effectively promote conceptual understanding of the limit concept when coupled with a mature conceptual model of development. Four themes influencing development of these understandings emerged: An instructional focus on skills over concepts, the instructional sequence, the willingness and ability of students to adopt and utilize computational tools, and the ways cognitive conflict was mediated.

ASSESSING THE IMPACT OF COMPUTER PROGRAMMING
IN UNDERSTANDING LIMITS AND DERIVATIVES
IN A SECONDARY MATHEMATICS
CLASSROOM

by

Christopher H. de Castro

A Dissertation

Presented in Partial Fulfillment of Requirements for the
Degree of
Doctor of Philosophy
in
Teaching and Learning - Mathematics Education
in
the Department of Middle-Secondary Education and Instructional Technology
in
the College of Education
Georgia State University

Atlanta, GA
2011

TABLE OF CONTENTS

List of Tables……………………………………………………………………………...v
List of Figures……………………………………………………………………………vi
Abbreviations…………………………………………………………………….....xix

Chapter Page

1 INTRODUCTION 1

Background of Study 1
Problem Statement 4
Significance of the Problem 7
Guiding Questions 10
Theoretical Basis for Study..11
Constructionist Learning...11
Action-Process-Object-Schema Theory....................................13
Methodology/Approaches..14

2 LITERATURE REVIEW 16

Introduction..16
Programming..16
Computer Algebra Systems 22
Student Understanding of the Limit Concept 29
Student Understanding of the Derivative Concept 34
Programming and Calculus..35
Conclusion 36

3 METHODOLOGY 37

Qualitative Methodology 38
Case-study Methods...39
Data Sources 41
Research environment..42
Research Participants 43
Case P: Programming Groups...46
Case N: Non-programming Groups 48
Role of the Researcher 50
Role of the Conceptual Model 55
Key Assumptions in Conceptual Model 60
APOS Decompositions 60
Data Collection Procedures and Instrumentation 63
Procedures..63
Document Artifacts..68
Observations 70
Reflexive Researcher Journal 70
Role of Computer Software 70
Data Analysis...71
Document Analysis..72

Chapter Page

Trustworthiness of Results....73
Limitations....76
Summary....76

4 RESULTS....78

Instructional Context....79
Initial Survey....83
Initial Limit Conception....84
Lab 1 Results....85
Group P_1....86
Group P_2....89
Group P_3 (singleton)....93
Group P_4 (singleton)....97
Group N_1....99
Group N_2....101
Group N_3....103
Group N_4....104
Lab 1 Summary....108
Lab 2 Results....110
Group P_1....112
Group P_2....118
Group P_3 (singleton)....124
Group P_4 (singleton)....132
Group N_1....138
Group N_2....143
Group N_3....150
Group N_4....156
Lab 2 Summary....162
Lab 3 Results....164
Group P_1....165
Group P_2....175
Group P_3....183
Group P_4....194
Group N_1....201
Group N_2....212
Group N_3....223
Group N_4....232

Chapter Page

Tool Use and Justification in Lab 3 241
Lab 3 Summary....242
Conceptions of Limit Following Lab 3....245
Lab 4 Results....246
Group P_1 248
Group P_2 253
Group P_3 (singleton)....254
Group P_4 (singleton)....254
Group N_1 255
Group N_2 262
Group N_3 270
Group N_4 279
Lab 4 Summary....287
Final Limit Conceptions 288

5 CONCLUSIONS....289

Conceptions of the Limit Concept 290
The Programming Groups Final Conception of Limit....291
The Non-programming Groups Final Conception of Limit....291
Variations in Limit Concept....291
Levels of APOS Attainment 297
Sources of Differences in Conceptual Understanding 299
The Role of the Instructional Focus....299
The Role of the Instructional Sequence 302
The Role of Computational Tool Adoption 303
The Role of Pair-Programming....304
Implementation Challenges 306
Future Research Possibilities 308
Conclusions....309

REFERENCES 311

APPENDIXES....323

LIST OF TABLES

Table Page

1 Demographics of Research Site……………………………………………..42

2 Data Collection Summary…………………………………………………..68

LIST OF FIGURES

Figure Page

1 Programming Utilizes and Unifies Varied Mathematical Concepts 5

2 A Sample Session in the Maple Computer Algebra System 24

3 Units of Analysis in the Proposed Multiple Case Embedded Design 40

4 Participant parings based upon gender and prior mathematical performance.. 46

5 The Role of a Conceptual Model 56

6 The Conceptual Model 57

7 APOS Decomposition for Understanding the Limit 61

8 APOS Decomposition for Understanding the Derivative 62

9 Overview of Data Sources.. 63

10 The formal definition of Limit 80

11 Formulations of function definition. The non-programming groups (left) utilized a traditional mapping formulation and the programming groups (right) utilized a procedural formulation of function definition. 82

12 Initial Conception of Limit 84

13 Group dialog with Maple CAS for Group P_1, Lab 1, Problem 1(a,b,d)........... 86

14 Group P_1's response to Lab 1, Problem 1(e)... 87

15 Group P_1's response to Lab 1, Problem 1(f) 87

16 Group P_1's response to Lab 1, Problem 3(d,e).. 88

17 Group P_1's response to Lab 1, Problem 4(a-d) 88

18 Group P_2's response to Lab 1, Problem 1(b) 89

19 Group P_2's response to Lab 1, Problem 1(f).. 90

Figure Page

20 Group P_2's response to Lab 1, Problem 3(d-g) ... 91

21 Group P_2's response to Lab 1, Problems 1(e) and 4(d) ... 92

22 Group P_3's response to Lab 1, Problem 1(a,b) ... 94

23 Group P_3's response to Lab 1, Problem 1(d) ... 94

24 Group P_3's response to Lab 1, Problem 1(f) ... 94

25 Group P_3's response to Lab 1, Problem 3(d,e) ... 95

26 Group P_3's response to Lab 1, Problem 3(g) ... 96

27 Group P_4's response to Lab 1, Problem 1(f) ... 98

28 Group P_4's response to Lab 1, Problem 3(d,e) ... 98

29 Group P_4's response to Lab 1, Problem 4d ... 98

30 Group N_1's response to Lab 1, Problem 1(a,b) ... 100

31 Group N_1's response to Lab 1, Problem 1(e,f) ... 100

32 Group N_1's response to Lab 1, Problem 3(d,e,g) ... 100

33 Group N_2's response to Lab 1, Problem 1(b,f) ... 102

34 Group N_2's response to Lab 1, Problem 3(b-d) ... 102

35 Group N_3's response to Lab 1, Problem 1(e-f) ... 104

36 Group N_4's response to Lab 1, Problem 1(a,b,d) ... 105

37 Group N_4's response to Lab 1, Problem 2d ... 105

38 Group N_4's response to Lab 1, Problem 4(a) ... 106

39 Group N_4's response to Lab 1, Problem 1(e-f) ... 106

Figure Page

40 Group N_4's response to Lab 1, Problem 2(e) 106

41 Group N_4's response to Lab 1, Problem 3(b-e) 106

42 Group N_4's response to Lab 1, Problems 4(d)... 107

43 APOS Level of Attainment Following Lab 1. The shaded region represents the actual attainment and the vertical line indicates the intended attainment level.. 109

44 Mystery Functions for Exploration by all groups in Lab 2 110

45 Sample Application of `simpleLimitTable` tool 111

46 Group P_1's response to Lab 2, Problems 1-3 exploring function `f(x)` 113

47 Group P_1's `SimpleLimitTable` Implementation and Demonstration 113

48 Groups P_1's exploration of function g(x) using the `simpleLimitTable` procedure 114

49 Group P_1's exploration of m(x) near x=2 to a tolerance of 0.001 115

50 Group P_1's Implementation of `simpleLimitTable2` and application at x=2 115

51 Group P_1's implementation of `simpleLimitTable3` near x=2 116

52 Group P_1's response to Lab 2, Problem 8.. 117

53 Group P_2's response to Lab 2, Problem 1-3 exploring mystery function f(x) 118

54 Group P_2's `simpleLimitTable` implementation and demonstration....... 119

55 Group P_2's response to Lab 2, Problem 1 exploring mystery function g(x) ... 119

Figure Page

56 Group P_2's exploration of mystery function m(x) near x=2 with a tolerance of 0.001 ... 120

57 Group P_2's implementation of `simpleLimitTable2` ... 121

58 Group P_2's implementation and demonstration of `simpleLimitTable3` ... 122

59 Group P_2's response to Lab 2, Problem 8 exploring mystery function m(x) .. 122

60 Group P_3's implementation of `simpleLimitTable` ... 124

61 Group P_3's response to Lab 2, Problem 1 exploring mystery function g(x) ... 125

62 Group P_3's response to Lab 2, Problem 2 exploring mystery function h(x) ... 126

63 Group P_3's response to Lab 2, Problem 2 exploring mystery function k(x) ... 126

64 Group P_3's exploration of mystery function m(x) with tolerance 0.001 ... 127

65 Group P_3's misunderstanding of formal versus actual parameters ... 128

66 Group P_3's implementation and demonstration `simpleLimitTable2` ... 128

67 Group P_3's implementation and demonstration of `simpleLimitTable3` ... 129

68 Group P_3's response to Lab 2, Problem 8 exploring mystery function m(x) .. 131

69 Group P_4's response to Lab 2, Problems 1-3 exploring mystery function f(x) ... 132

70 Group P_4's implementation of `simpleLimitTable` ... 133

71 Group P_4's response to Lab 2, Problem 1 exploring mystery function g(x) ... 134

Figure Page

72 Group P_4's exploration of mystery function m(x) near x=2

with tolerance 0.001 134

73 Group P_4's implementation of demonstration of `simpleLimitTable2`

on mystery function m(x) near x=2 with tolerance 0.0001 135

74 Group P_4's implementation of demonstration of `simpleLimitTable3`

on mystery function m(x) near x=2 with tolerance 0.00001 136

75 Group P_4's response to Lab 2, Problem 8 exploring mystery function m(x) .. 137

76 Group N_1's response to Lab 2, Problems 1-3 exploring mystery function

f(x) 138

77 Group N_1's response to Lab 2, Problems 1-3 exploring mystery functions

g(x), h(x), and k(x) 139

78 Group N_1's response to Lab 2, Problems 2-6 exploring mystery function

m(x) near x=2 140

79 Group N_1's response to Lab 2, Problem 8 exploring mystery function

m(x)

near x=2 142

80 Group N_2's response to Lab 2, Problems 1-3 exploring mystery function

f(x) 144

81 Group N_2's response to Lab 2, Problem 1 exploring mystery function

g(x) near x=1 144

82 Group N_2's conclusions relating to mystery functions g(x) and h(x) 145

Figure Page

83 Group N_2's response to Lab 2, Problems 2-6 exploring mystery function m(x) near x=2 ... 147

84 Group N_2's response to Lab 2, Problems 7-8, Exploring mystery function m(x) near x=2 ... 148

85 Group N_3's response to Lab 2, Problems 1-3, Exploring mystery function f(x) ... 150

86 Group N_3's response to Lab 2, Problem 1, Exploring mystery function g(x) ... 151

87 Group N_3's response to Lab 2, Problem 2, Exploring mystery function h(x) ... 152

88 Group N_3's response to Lab 2, Problem 3, Exploring mystery function k(x) ... 152

89 Group N_3's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerance 0.001 ... 153

90 Group N_3's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerances 0.1, 0.01, 0.001, 0.0001 and 0.00001 ... 154

91 Group N_3's response to Lab 2, Problem 8, Exploring mystery function m(x) near x=2 ... 155

92 Group N_4's analysis of mystery function f(x) ... 156

Figure Page

93 Group N_4's analysis of mystery function g(x) 157

94 Group N_4's analysis of mystery function h(x) 158

95 Group N_4's analysis of mystery function k(x) 158

96 Group N_4's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerance 0.01 159

97 Group N_4's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerances 0.1, 0.01, 0.001, 0.0001 and 0.00001 160

98 Group N_4's response to Lab 2, Problem 8, exploring mystery function m(x) near x=2 161

99 APOS Level of Attainment post Lab 2. Shaded region represents actual attainment, vertical bar indicates the intended level of attainment. 163

100 Group P_1's Definition of `leftLim` and `rightLim` procedures in Lab 3 ... 165

101 Group P_1's Analysis of mystery function f(x) 167

102 Group P_1's analysis of mystery function g(x) 168

103 Group P_1's analysis of mystery function h(x) 169

104 Group P_1's analysis of mystery function k(x) 170

105 Group P_1's constructed function b(x) for Lab 3 170

106 Group P_1's constructed function c(x) in Lab 3 171

107 Group P_1's constructed function d(x) in Lab 3 172

108 Group P_1's constructed function e(x) in Lab 3 173

Figure Page

109 Group P_1's response to Post-lab 3 questions 174

110 Group P_2's Definition of `leftLim` and `rightLim` procedures in Lab 3.... 175

111 Group P_2's analysis of mystery function f(x) 176

112 Group P_2's analysis of mystery function g(x) in Lab 3 177

113 Group P_2's analysis of mystery function h(x) in Lab 3 178

114 Group P_2's analysis of mystery function k(x) in Lab 3 179

115 Group P_2's constructed function b(x) in Lab 3 180

116 Group P_2's construction of function c(x) in Lab 3 181

117 Group P_2's construction of function d(x) in Lab 3 182

118 Group P_2's construction of function e(x) in Lab 3 182

119 Group P_3's implementation of `leftLim` and `rightLim` procedures 183

120 Group P_3's analysis of mystery function f(x) in Lab 3 184

121 Group P_3's analysis of mystery function g(x) in Lab 3 186

122 Group P_3's analysis of mystery function h(x) in Lab 3 187

123 Group P_3's analysis of mystery function k(x) in Lab 3 188

124 Group P_3's construction of function b(x) in Lab 3 189

125 Group P_3's construction of function c(x) in Lab 3 190

126 Group P_3's construction of function d(x) in Lab 3 191

127 Group P_3's construction of function e(x) in Lab 3 192

128 Group P_3's response to Post-lab 3 questions 193

Figure Page

129 Group P_4's implementation of `leftLim` and `rightLim` procedures 194

130 Group P_4's attempt to redefine mystery function f(x) in Lab 3 195

131 Group P_4's analysis of mystery function f(x) in Lab 3 196

132 Group P_4's construction of function b(x) in Lab 3... 197

133 Group P_4's construction of function c(x) in Lab 3 197

134 Group P_4's construction of function d(x) in Lab 3... 198

135 Group P_4's construction of function d(x) in Lab 3... 199

136 Group P_4's Selected Responses to Post-lab 3.. 200

137 Group N_1's analysis of mystery function f(x) in Lab 3.................................. 202

138 Group N_1's analysis of mystery function g(x) in Lab 3 203

139 Group N_1's analysis of mystery function h(x) in Lab 3 204

140 Group N_1's analysis of mystery function k(x) in Lab 3 205

141 Group N_1's construction of function b(x) in Lab 3 206

142 Group N_1's construction of function c(x) in Lab 3 207

143 Group N_1's construction of function d(x) in Lab 3 208

144 Group N_1's construction of function e(x) in Lab 3 209

145 Group N_1's Selected Responses to Post-lab 3.. 210

146 Group N_2 's analysis of function f(x) in Lab 3 212

147 Group N_2 's analysis of mystery function g(x) in Lab 3 214

148 Group N_2 's analysis of mystery function h(x) in Lab 3 215

Figure Page

149 Group N_2 's analysis of mystery function k(x) in Lab 3 216

150 Group N_2 's construction of function b(x) in Lab 3 217

151 Group N_2 's construction of function c(x) in Lab 3....................................... 218

152 Group N_2 's construction of function d(x) in Lab 3 219

153 Group N_2 's construction of function e(x) in Lab 3....................................... 220

154 Group N_2 's Selected Response from Post-lab 3 221

155 Group N_3 's analysis of mystery function f(x) in Lab 3 223

156 Group N_3 's analysis of mystery function g(x) in Lab 3................................. 224

157 Group N_3 's analysis of mystery function h(x) in Lab 3................................. 225

158 Group N_3 's analysis of mystery function k(x) in Lab 3................................. 226

159 Group N_3 's construction of function b(x) in Lab 3....................................... 227

160 Group N_3 's construction of function c(x) in Lab 3 228

161 Group N_3 's construction of function d(x) in Lab 3....................................... 229

162 Group N_3 's construction of function e(x) in Lab 3 230

163 Group N_3 's Selected Response from Post-lab 3 231

164 Group N_4 's analysis of mystery function f(x) in Lab 3 232

165 Group N_4 's analysis of mystery function g(x) in Lab 3 233

166 Group N_4 's analysis of mystery function h(x) in Lab 3 234

167 Group N_4 's analysis of mystery function k(x) in Lab 3 235

168 Group N_4 's construction of function b(x) in Lab 3 236

Figure Page

169 Group N_4's construction of function c(x) in Lab 3. 237

170 Group N_4's construction of function d(x) in Lab 3 238

171 Group N_4's construction of function e(x) in Lab 3 239

172 Group N_4's post-lab responses to Lab 3 240

173 Tool Usage in Lab 3. The upper bar indicates the number of characteristics successfully created. The lower bar indicates the number of successful justifications using the `leftLim` and `rightLim` tools. 241

174 APOS Level of Attainment following Lab 3. The shaded region represents actual attainment and the vertical line indicates the intended level of attainment. 244

175 Postlab 3 Conceptions of Limit 245

176 The `dePlot` tool for exploring the formal $\varepsilon-\delta$ definition of limit. 246

177 Group P_1's implementation of the `dePlot` tool. 249

178 Group P_1's response to Lab 4, Problems 1 and 2. 250

179 Group P_1's response to Lab 4, Problem 3 251

180 Group P_1's response to Lab 4, Problem 5. 251

181 Group P_2's implementation of `dePlot` tool. 253

182 Group N_1's response to Lab 4, Problem 1. 255

183 Group N_1's response to Lab 4, Problem 2 256

184 Group N_1's response to Lab 4, Problem 3 257

185 Group N_1's response to Lab 4, Problem 4 258

Figure Page

186 Group N_1's response to Lab 4, Problem 5. 259

187 Group N_1's response to Lab 4, Problems 6 and 7 260

188 Group N_1's understanding of connection between columns of limit tables and rectangles in dePlots 261

189 Group N_2's response to Lab 4, Problem 1 262

190 Group N_2's response to Lab 4, Problem 2 263

191 Group N_2's response to Lab 4, Problem 3 264

192 Group N_2's response to Lab 4, Problem 4 265

193 Group N_2's response to Lab 4, Problem 5 266

194 Group N_2's response to Lab 4, Problems 6a and 7. 267

195 Group N_2's response to Lab 4, Problems 6b and 7 268

196 Group N_2's understanding of connection between columns of limit tables and rectangles in `dePlots` 269

197 Group N_3's response to Lab 4, Problem 1 270

198 Group N_3's response to Lab 4, Problem 2 271

199 Group N_3's response to Lab 4, Problem 3 272

200 Group N_3's response to Lab 4, Problem 4 273

201 Group N_3's response to Lab 4, Problem 5 274

202 Group N_3's response to Lab 4, Problem 6a and 7 275

203 Group N_3's response to Lab 4, Problem 6b and 7 276

204 Group N_3's response to Lab 4, Problem 6c and 7 277

Figure		Page

205 Group N_3's members understanding of the connection between the columns in limit tables and the rectangles in `dePlots`s 278

206 Group N_4's response to Lab 4, Problem 1 279

207 Group N_4's response to Lab 4, Problem 2 280

208 Group N_4's response to Lab 4, Problem 3 281

209 Group N_4's response to Lab 4, Problem 4 282

210 Group N_4's response to Lab 4, Problem 5 283

211 Group N_4's response to Lab 4, Problem 6a, 7 284

212 Group N_4's response to Lab 4, Problem 6b, 7 285

213 Group N_4's understanding of connection between columns of limit tables and rectangles in `dePlots` 286

214 Final APOS Level of Attainment. The shaded region represents actual attainment and the vertical line indicates the intended level of attainment. 288

215 Final Conceptions of Limit 288

216 Changes in Limit Conception for all Groups 290

217 Final APOS Level of Attainment. The shaded region represents actual attainment and the vertical line indicates the intended level of attainment. 297

ABBREVIATIONS

APOS	Action-Processes-Objects-Schema
CAS	Computer Algebra System
NCTM	National Council of Teachers of Mathematics

CHAPTER ONE

INTRODUCTION

Background of Study

The potential use of computer programming as an instrument of mathematical learning has been repeatedly promoted since the early 1970's (Clements, 1999; Feurzeig, Papert, Bloom, Grant, & Solomon, 1970; D. C. Johnson, 2000; Sfard & Leron, 1996; Soloway, 1993; Sylvia, 1986; Thomas & Upah, 1996). Results into the effectiveness of programming as an instructional tool in mathematics education have been mixed, as will be discussed, suggesting mathematics educators lack fundamental understanding of programming's potential. This lack of understanding, coupled with routine use of computing environments in mathematics classrooms (Crowe & Zand, 2001; Demana & Waits, 2000), current curricular trends vigorously promoting the use of technological tools (NCTM, 2003), and more mature coherent instructional design methodologies (Asiala et al., 1996; Clements, 1999; Clements & Sarama, 1995; Harel & Papert, 1991; Perkins & Salomon, 1992) strongly suggests revisiting programming as a potential cognitive tool in the mathematics classroom.

Computational tools such as graphing calculators (Demana & Waits, 2000; Graham, 2003; Guin & Trouche, 1999; Jones, 2005), the Geometer's Sketchpad, and computer algebra systems (CAS) such as Maple and Mathematica are utilized routinely to provide students with alternative representations and more realistic models by removing

the computational drudgery of complex symbolic manipulations. Students rapidly produce alternative representations such as colorful graphs and tables and discover trends and patterns that are impractical to pursue with paper-and-pencil. There has been a great deal of research focused on providing multiple and multiple-linked representations (Mor, Hoyles, Kahn, Noss, & Simpson, 2004; Parnafes & Disessa, 2004; Ploger, 1991; Ploger & Carlock, 1996; Sherin, 2001; Simpson, Hoyles, & Noss, 2005) and visualizations (Crowe & Zand, 2000; Zimmermann, 1991).

Sfard (1991) suggested a dual nature to mathematical understanding- *structural* and *procedural*. While mathematical objects generally are not observable with our senses, structural knowledge corresponds with concrete ways of representing abstract ideas; i.e. a number written on a page of paper or a geometrical figure depicting symmetry. Procedural knowledge is a dynamic knowledge that views objects as potentialities – things that might come into existence as a result of some process, i.e. the process of adding two numbers. It is through a complex interplay between these two modes of exploration and understanding that mathematical learning arises.

The ability of producing multiple representations and visualizations of mathematical concepts provides strong support for the development of structural understanding but does little to promote procedural or operational understanding. These permit students greater potential to "see mathematics" but do not necessarily engage them in "doing mathematics." (Stein, Smith, Henningsen, & Silver, 2000) This use of visualization, while extremely useful, fails to capitalize on the operational perspective also offered by the technology thus limiting its potential as a cognitive tool. For example, spreadsheets, like Excel, and graphing calculators can produce tables and graphs

visualizing trends from which a student might understand the mathematical concept of a limit. While this perspective may give the student the means of discerning a limiting value, it does not prompt reflection upon his/her unconscious internal processes used to *find* the limit. With the limit concept (and many other calculus concepts), there are two ways to view the limit – as an object (a static entity; a number) and as a process producing an object.

For example, consider the mathematical expression: $2+3$. This expression can be understood in two very different ways. On the one hand, $2+3$ can be thought of as the *value* resulting from the addition, 5. On the other hand, the expression may also be understood as a prompt to perform the *process* of addition of the numbers 2 and 3. Gray and Tall (1994) refer to such concepts understandable both as a process producing an object as well as the object produced as *procepts*.

The utilization of computer algebra systems strictly for visualization overlooks the procedural view of mathematical understanding that plays a crucial role in developing deep understanding of "proceptual" mathematical concepts like the limit and derivative. Calculus students have routinely found graphical visualizations as a means for understanding more intellectually challenging than performing traditional algorithmic processes such as differentiation and integration (Berry & Nyman, 2003; Habre & Abboud, 2006; Orton, 1983). In producing the visualizations for themselves, students are doing mathematics rather than passively receiving a visual explanation of a mathematical concept. Current symbolic algebra systems, such as Maple and Mathematica, not only permit students to easily produce alternative representations they also provide a programming interface to construct more meaningful interactions within the specific

domain of mathematics. Programming interfaces permit direct manipulation of varied representations providing the learner with a means of *interacting* with multiple representations. Eisenberg (1995) argued for such extension of visual environments with programming language constructs. Interest in such representational interactions is reflected in recent Microworld studies on Boxer (A. DiSessa, 2000; A. A. DiSessa, Abelson, & Ploger, 1991), StarLogo, etc..

Jonassen, Carr, & Yueh (1998) argued that technologies should not support learning by attempting to instruct the learners, but rather should be used as knowledge construction tools that students learn *with*, not *from*. In this way, learners function as designers, and the computers function as Mindtools for interpreting and organizing their personal knowledge.

Problem Statement

This research explored the development of student understanding of two key calculus concepts, the limit and the derivative, through the development and/or use of programming-based visualization tools in the Maple computer algebra system. The goal of the research was to explore, characterize and better understand the development of these mathematical concepts when learners constructing and utilizing software tools were contrasted with learners utilizing pre-constructed non-programming based CAS visualization tools.

My prior experience teaching computer science led to the observation that, in developing software, students develop deeper mathematical understanding. The act of writing a computer program promotes active engagement with key mathematical ideas.

For example, consider the act of writing a computer program to perform the familiar mathematical operation of adding two fractions. This task is outlined in *Figure* 1. In constructing such a program, the student must address several important questions. How does one represent a fraction within the computer? How does one add two fractions? How can you tell if a fraction can be reduced? How do you reduce a fraction? In addressing these questions, the student in actually engaged with a specific mathematical concept, the fraction, as they attempt to construct a working program. Also shown in the figure are necessary programming constructs and mathematical concepts that necessarily enter in to the program design process.

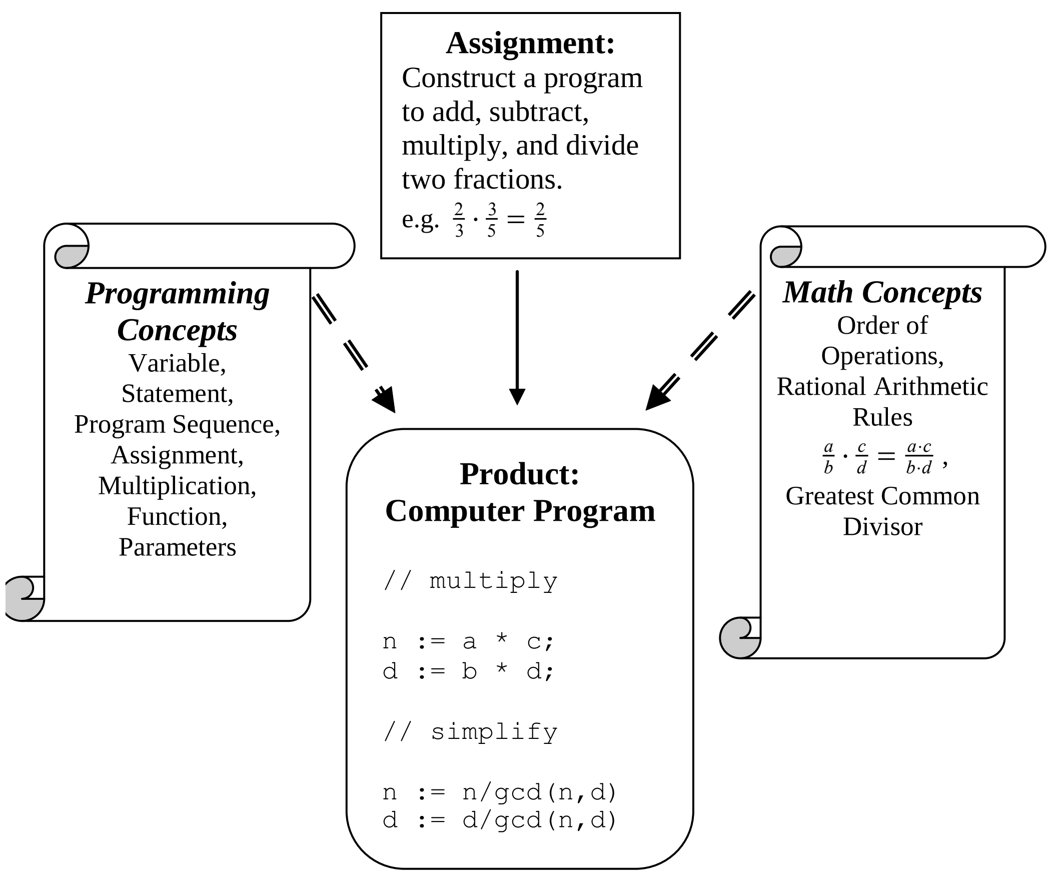


Figure 1. Programming Utilizes and Unifies Varied Mathematical Concepts

Computer algebra systems are becoming quite common in mathematics courses at the post-secondary level (Crowe & Zand, 2001; Fey, Cuoco, Kieran, McMullin, & Zbiek, 2003; Heid, 1984, 1988; Heid, Blume, Hollebrands, & Piez, 2002; Heid & Edwards, 2001). In their most common applications, CAS's provide students with the ability to quickly perform algebraic and graphical operations that are tedious and time-consuming to perform by hand. Additionally, they permit deeper exploration of mathematical ideas by providing convenient access to multiple varied representations (i.e. graphs, tables, equations). However, most computer algebra systems also provide programming constructs and development environments that remain largely unutilized. Such environments offer opportunities for students to not only see alternative representations but also to create them, interact and learn with them. These interactions promote reflection, integration, and unification of mathematical concepts. Students not only see varied representations by see how representations are related and how they can be used in concert to explore mathematical concepts.

Calculus, being the mathematics of change, is dynamic. Many key ideas are related to processes that are easily realizable using programming constructs. Students can implement algorithms that develop structural and procedural understanding of previously static mathematical notions. Noss (1997) states that mathematical meaning can come from an individual's awareness that a particular expression can be recognized by a computer. The process of entering statements in a programming language, testing them and using them induces one to construct meaning from these statements. Programming promotes direct active engagement with mathematical concepts.

In this study, we desired to move beyond the utilization of visualizations to consider the implementation of processes developing specific mathematical concepts, such as the limit. In what ways were students' conceptual understandings the same (different) having constructed computer programs? Has the conceptual understanding of the underlying associated mathematical structures deepened? In what ways?

A recurring theme in research relating to the effectiveness of computer programming has been the importance of appropriate instructional design: The importance of teacher, designer, and researcher in structuring computer-based activities and the role of the teacher in guiding the activities that lead to constructivist learning. To address this concern, CAS activities, both programming-based and non-programming based, were developed within two coherent theoretical frameworks: the action-process-object-schema (APOS) framework for instructional design and constructionist learning theory.

Significance of the Problem

The National Council of Teachers of Mathematics' (NCTM) Principles and Standards for School Mathematics (NCTM, 2003), provides guidelines for excellence in mathematics education and issues a call for all students to engage in more challenging mathematics. These standards serve as a unifying framework for mathematics education by presenting a more comprehensive picture of the "whats" and "hows" of mathematics education. The underlying philosophy is constructivism, which asserts that students learn mathematics by active involvement with mathematical models that allow them to internally construct understandings and concepts. This means a decrease in the amount of drill and practice in any medium and increased interaction with a variety of models of

mathematical concepts. Computational skill is deemphasized and use of calculators and computers is encouraged. Computers are extremely important because they can provide a variety of rich experiences that allow students to be more actively involved with mathematics.

The NCTM Standards consist of ten standards, five *content* standards, and five *process* standards. Programming closely aligns with all five process standards: Problem Solving, Reasoning and Proof, Communication, Connections, and Representation. Programming involves mathematical exploration, the *development* and *use*, rather than mere *use*, of algorithmic thinking, the development of various representations, and construction of environments in which students have metaphors and models of various modes of thinking that extend their range of solution strategies.

Consider the act of writing a computer program. A learner must:

1. analyze a problem statement, typically stated as a word problem and express its essence, abstractly and with examples; (NCTM Standard 1)
2. formulate statements and comments in a precise language; (NCTM Standards 3 and 5)
3. evaluate and revise these activities in light of checks and tests (NCTM Standard 2); and pay attention to details.

Programming is an active task consistent with constructivist methodology that extends traditional instructional activities in a manner that promotes higher-level thought *processes*. Programming helps students learn to mediate their problem solving processes in requiring the finding and correcting of flaws in reasoning, a process known as *debugging*.

In the past, students have inhabited educational environments where the objective was to "get the correct answer." Programming promotes reflectiveness in problem solving rather than a binary outcome from a sequence of prescribed steps or a turn of a mystical mathematical crank. The presence of a programming environment changes the very character of the mathematics classroom from one in which students are conditioned to avoid mistakes to one in which mistakes lead to reflection and experimentation (Sfard & Leron, 1996).

Programming also engenders construction of mental representations which Davis and Maher (1997) suggest is a key mechanism for obtaining new mathematical meaning. Ploger and Carlock observed the following:

> The act of programming forces students to choose what to represent for a particular program, encouraging them to abstract the problem from the particular situation. Consequently, they pay close attention to details on the program representation. To complete their work, they must not only observe the representations they must actively create them. This study indicates that programming can be useful in helping students organize their knowledge of a complex process and focus on relevant information. (Ploger & Carlock, 1996)

Studies have also shown that the study of computer programming is intellectually rewarding for young children in elementary school, and for computer science majors in college (Lawler, 1986) yet broad integration of programming into the classroom has not been vigorously pursued. It has not been clear how programming relates to specific skills-based curricula, making it difficult for educators to see any fit for programming (A.

DiSessa, Hoyles, Noss, & Edwards, 1995; Ioannidou, Repenning, Lewis, Cherry, & Rader, 2003). Further, research suggests that becoming a proficient programmer takes years and that programming itself might be as difficult or more difficult to master than the mathematical material itself (Kennedy, 2002). NCTM standards are shifting curricular goals away from mere skill-development towards a curriculum stressing both *content* and *process* in mathematical understanding. With the advent of domain specific programmable applications, such as CAS in mathematics education, there is not the need for students to learn a general-purpose programming language; students need not become experienced programmers in order to derive meaning and understanding from programming.

Guiding Questions

In this study, two in-tact classes of Advanced Placement Calculus AB students explored the concepts of limit and derivative from differential calculus using exploratory tools in a computer algebra system. One class, P, comprised a group of students who utilized programming to construct exploratory tools and subsequently explore the limit and derivative concepts. The other class, N, comprised a group of non-programming students who performed similar explorations using pre-constructed visualization tools. This research explored the following questions.

(1) How does the development of conceptual understanding of limit and derivative contrast between students constructing and utilizing programming based exploratory tools as compared with students utilizing preconstructed exploratory tools in a CAS environment?

(2) What factors influence these two developmental trajectories?

Theoretical Basis for Study

As much previous programming research suggests, computer programming is most effective when aligned with specific instructional goals (Clements, 1999; Linn & Dalbey, 1985). Learning in unstructured computing environments may suffer from unreflective tool use and avoidance of mathematical analysis (Noss & Hoyles, 1992). There is need for a framework for the development of meaningful and relevant programming activities supporting the development of the limit and derivative concepts. Two theoretical frameworks fundamentally supported this study: *Constructionist Learning* and the *Action-Process-Object-Schema* (APOS) framework for research in undergraduate mathematics education.

Constructionist Learning

Constructionist learning is an epistemological perspective, due to Harel and Papert (1991), consonant with contructivist principles that learning is an active process in which individuals build understanding by constructing knowledge structures. Constructionism adds that this knowledge construction is well facilitated through the construction of realizable products.

Constructionist learning more directly addresses the ways in which individuals interact with their constructions, how they support self-directed learning, and ultimately facilitate knowledge construction. In this study, computer programs will be seen as the realized products and the computing environment as the medium through which learners interact with their constructions.

Within this framework, Harel (1991) engaged 17 fourth grade students as software designers in their Instructional Software Design Project (ISDP). The goal of the project was for students to learn basic rational-number arithmetic and Logo programming through the implementation of a long-term software design activity. Could software design serve as a learning tool that addresses learning and conceptual understanding in several domains at once, in this case, learning Logo program, and learning rational arithmetic?

Using an integrated approach the researcher taught relevant rational arithmetic operations and basic Logo programming via the use of Logo programming while students were charged with the development of a long-term instructional design project to teach fractions. Students worked on this project for one hour a day, four times a week, for four months. The researcher additionally took research notes, compiled students Designer Notebooks that were logs written and edited by the students, and empirically assessed understanding.

The design project required thinking skills such as self-management, reflection, planning, revising, and representing. Students used Logowriter, a Logo variant, to create instructional software on the topic of fractions for other students to use. As a result, students improved their ability to work with fractions and learned more about *Logo* programming than two control groups. The results also showed that students developed enhanced metaconceptual and metacognitive awareness; they acquired cognitive flexibility, control over solution processes, and greater self-confidence in problem-solving. In (Harel, 1991), she further identified tendencies of Logo-based programming

to allow for individual variations in "learning, mastery, and self-expression" in children, and called for further research into the nature of these differences.

Action-Process-Object-Schema Theory

The Action-Process-Object-Schema (APOS) framework provides a framework for the development of instructional activities targeted at specific curricular outcomes (Asiala, et al., 1996). The essential idea behind APOS theory is evolutionary- that before students are able to understand particular mathematical concepts, they must first have a context or framework of mental constructions that permits the understanding of the concepts. APOS theory posits that learning certain mathematical concepts involves making specific mental constructions for use in understanding mathematical problems. Thus, a student is not able to learn a given concept until specific prior mental representations are constructed. The goal of APOS theory is to understand what these mental constructions are and then formulate a model of concept development that permits the construction of meaningful activities scaffolding the necessary mental constructions.

These ideas are similar to those of Sfard (1992) who suggested a similar notion of cognitive development in considering the historical and psychological transition of mathematical understanding from a process or *operational* conception to a static *structural* conception. She identified three transitional stages from an operational to a structural conception -- *interiorization*, *condensation*, and *reification*. Interiorization occurs when a person can step through a particular process. Condensation occurs when the person views a process as a whole and may utilize it as a sub-process in other processes. Reification occurs when the process may be viewed structurally as an object. Processes are operations on previously established objects. Each process is reified into an

object to be acted on by other processes. This forms a chain of process-object transitions. APOS Theory possesses a similar theoretical perspective but provides a framework for driving these developmental transitions.

A meta-analysis of 13 studies involving calculus, abstract algebra, functions, quantification, induction, and affective domain suggest that APOS theory is an effective tool form assisting students in learning mathematics (Weller et al., 2000).

Methodology/Approaches

In this study, two groups of secondary students in Advanced Placement Calculus AB classes were contrasted. Results of two prior APOS studies related to understanding the limit and derivative (Asiala, Cottrill, Dubinsky, & Schwingendorf, 1997) and the constructionist framework were utilized to frame the construction of appropriate computer programming activities. Students utilizing a programming based approach programmed exploratory tools in pairs and subsequently used them to demonstrate and explore the limit and derivative concepts. Students utilizing a non-programming based approach performed corresponding explorations in pairs using preconstructed tools provided to them. Neither group was assumed to have any prior programming experience.

Previous APOS studies evaluated student performance utilizing interviews in which mathematical questions were posed during an interview and previous examination results were examined (Asiala, et al., 1996; Asiala, et al., 1997; Cotrill et al., 1996; Weller, et al., 2000). In this study, I assessed conceptual understanding qualitatively by seeking specific themes outlined by APOS decompositions as well as by eliciting notions of limit from students using common notions of limit and derivative derived in prior studies. My assessments were derived using the following data sources: (a) Examination

of written responses to probing questions in laboratory activities, (b) electronically recorded interactions within the CAS environment, (c) classroom observations, and (d) student responses to post-lab activities. Electronic documents were analyzed thematically to explore learning trends and themes suggested by the respective APOS decompositions as they relate to the instructional frameworks and the learner's conceptual understanding.

Dubinsky claimed students often construct meaning from formal symbolic systems (Dubinsky, 2000) suggesting that the symbol system utilized may promote and shape understanding in specific and unique ways. Sherin (2001) considered replacing standard algebraic notation in the physics classroom with a programming language. Sherin found that algebra-physics could be characterized as a physics of balance (i.e. static) whereas programming-physics can be characterized as a physics of processes and causation (i.e. dynamic); students conceptualize physics concept differently when a different expressive medium is utilized. This study presented a characterization of the nature of the understanding that develops when exploratory tools are developed and utilized via programming. Qualitative content analysis permitted contrasting of student's understandings and their developmental paths.

CHAPTER TWO

LITERATURE REVIEW

Introduction

This chapter presents a selected review of the research on computer programming, the use of computer algebra systems in mathematics education, and student understanding of the limit and derivative concepts. In structuring this literature review, foundational studies involving the application of computer programming to mathematics instruction will be examined leading to a discussion of how these studies transitioned research away from traditional programming towards more domain specific applications of computing technology. Next, the review will consider prior research into student understanding of the notions of limit and derivative then conclude with some research specifically related to the use of programming in calculus instruction. Throughout this review connections are drawn among these key foundational areas and how they inform the present study.

Programming

Computer programming has a long and rich research history (Clements & Meredith, 1993) with very mixed results. In 1970, Feurzeig et. al. (Feurzeig, et al., 1970) made four claims for the expected cognitive benefits of learning to program: Programming (a) provides justification for mathematical rigor, (b) encourages active mathematical exploration, (c) provides key insights into certain mathematical concepts,

and (d) provides a context for problem-solving and language with which to articulate their understandings.

Such claims prompted research into the question of whether learning to program had positive effects upon how students think and learn. Research in computer programming focused on the efficacy the programming language Logo to improve cognitive skills beyond the arena of programming and whether these skills transferred across domains. Pea and Kurland (1984) studied children exposed to 50 hours of programming instruction in Logo and found little evidence for transfer of planning or goal evaluation beyond the programming context. Kurland, Pea, Clement, and Mawby (1986) subsequently pursued this question with older students by considering the effectiveness of learning programming on problem solving skills of high school students. After two years of BASIC programming instruction, the authors found little evidence of improved problem-solving ability and little understanding of programming as well.

In a subsequent study, Pea, Kurland, and Hawkins (1987) studied elementary school children learning Logo in an attempt to answer the question of whether problem-solving skills developed in Logo transferred beyond programming activities. These studies suggested that with 30 hours of programming experience, there was not a significant difference between a control group and a Logo group on planning skills having little similarity to programming. Moreover, the authors found little evidence for transfer to near (similar) planning activities.

Although it appears that programming did not necessarily positively impact general thinking skills, there were positive results that provide some insight into how programming may be useful in mathematics education. Liao and Bright (1991) performed

a meta-analysis on the effects of computer programming on cognitive ability. They summarized 65 studies that compared 432 instances of programming versus non-programming groups with a weighted effect size of 0.41 for the programming groups. The meta-analysis indicated that Logo studies had significantly better results than BASIC and little difference than Pascal. This discrepancy was explained as a consequence of the differences between the structured programming languages like Logo and Pascal versus the unstructured BASIC environment. The authors further conclude that the programming experiences provide "a mildly effective approach for teaching students cognitive skills. "

Clements and Battista (1989, 1990) were instrumental in demonstrating the positive effectiveness of Logo programming on the acquisition of geometric understanding. The 1989 study of 24 fourth grade students presented with 78 lessons over a period of 26 weeks included a review of previous work, a teacher-centered presentation of new information, and independent student work on either teacher-assigned or self-selected problems. Students learned to write procedures and use variables in procedures. The experimental group was compared to a control group of 24 students who had similar mathematics achievement scores and who experienced a treatment of other types of computer programs, including writing, music, and drawing. The posttest was a researcher-developed structured interview measuring student knowledge of angles, angle measurement, shapes, and motion. The Logo group scored significantly higher than the control group on knowledge of angles, shapes, and motion. The researchers noted that the children in both groups maintained misconceptions about angles and figures, but that the Logo group had more experiences to draw on and were more likely to be able to construct and communicate the concepts. One key observation

was that a combination of teacher instruction and Logo programming was a crucial factor in the effectiveness of programming. An additional 1990 study confirmed these results with 12 additional students.

Logo proved to be effective in Geometry (Clements & Sarama, 1995, 1997) instruction but had more mixed results in other areas. As a general programming language, Logo provides many more expressive and powerful computing mechanisms, like lists, that went unutilized in these studies. Teachers concentrated almost exclusively on turtle geometry. Another key observation was that there existed a framework for understanding how geometric understanding develops in children, the Van Hiele model. Such understanding undoubtedly served as an instrumental instructional guide towards the development of effectively focused Logo activities.

In studies involving other programming languages, there were ambiguous results. One study (Blume & Schoen, 1988) showed no significant differences in problem solving effectiveness between BASIC programmers and non-programmers. In 1989, McCoy and Dodl (McCoy & Dodl, 1989) found a significant relationship between programming experience and mathematical problem-solving skill in 800 high school students. Such conflicting results might be explained by the difficulty in assessing problem-solving skill and the possible variety of programming experiences utilized. Such mixed results suggest that simple exposure to programming is not sufficient (Clements & Meredith, 1993).

Salomon and Perkins (1989) suggest that the such mixed results are explainable in terms of *transfer*. The authors argue that transfer occurs in two ways. *Low-road transfer,* depending on extensive varied practice, occurs by the automatic triggering of well-learned behavior in a new context and *high-road transfer* occurs through intentional

mindful abstraction of something from one context and application to a new context. The lack of low-road transfer in these programming studies can be explained by low levels of programming skill developed by students and the lack of high-road transfer due to instructional considerations. Instruction promoting mindful abstraction assuring thorough understanding of key abstractions is necessary to promote transfer. While (Pea & Kurland, 1984) provided direct instruction in programming, they did not connect this knowledge explicitly to the learning objective.

Difficulties associated with learning to program repeatedly occurred in studies. Pea and Kurland (1984) suggested a taxonomy of programming skills comprised of four distinct levels. At the first level, a *Program User* has an ability to utilize pre-packaged software such as games or demonstrations but has no understanding of how these programs accomplish their tasks. A *Code Generator* understands the syntax and semantics of a programming language, has an ability to read programs produced by others and can explain what each line of code accomplishes but is very limited in their ability to write complete programs. A *Program Generator* has mastered the syntax and semantics of the programming language, is so comfortable with the programming environment that they begin thinking about higher levels of design such as the use of subroutines. Finally, a *Software Developer* is able to write complete and useful programs that are intended to be used by others. Pea suggested that children learning to program can learn to write programs at the second level, *Program Generator*.

The ambiguity of such results suggested that, rather than altering the ways students think and learn, programming could serve to provide an additional cognitive tool that students could utilize in thinking and learning. More recently, studies involving

programming shifted from a psychological perspective of how students are affected by learning to program into more epistemologically founded views regarding how knowledge is constructed and how such constructions might be fostered.

This early work in programming with Logo, coupled with the difficulties of learning a programming language, suggested that there was a strong need for the instructional designer to identify boundaries for learning; that one should not focus entirely on the individual learner but upon providing a microworld that is "sufficiently bounded and transparent for constructive exploration and yet sufficiently rich for significant discover." (Papert, 1980, p. 208)

These difficulties prompted much of the current research into the area of Microworlds. Rieber (2003) defines a *microworld* as an interface between the learner and a software tool that is (a) domain-specific, (b) provides a doorway to the domain for the user by offer simple example within the domain that it immediately understandable to the learner, (c) leads to activity that can be intrinsically motivating to the learner, (d) leads to immersive activity best characterized as play or inquiry, and (e) is situated within the constructivist philosophy of learning. A key element to the success of the *microworld* approach is providing a specific mathematical context while at the same time providing an environment that is programming context-less.

Thus, there has been a trend away from the use of general purpose programming languages which are difficult to simultaneously master and related to specific mathematical learning objectives towards carefully constructed microworlds focused on specific mathematical content yet which provide programming constructs in a less syntax-bound environment (e.g. visual environments). It is hoped by the designers of

microworlds, that exploration of (and learning in) the environment is seamless – that actions within the environment are intuitively obvious to the student and using relevant programming constructs are understandable without explicit prior instruction.

In this research, we consider the use of a programming environment situated between these two extremes. A computer algebra system is a *microworld* uniquely designed for mathematical exploration yet which possesses all of the programming constructs found in general-purpose programming languages. Such a system straddles the two extremes and may provide potential learning opportunities when coupled with carefully designed instruction.

This previous research strongly suggests the ways that computers are utilized and aligned with instruction plays some role in their effectiveness. Technology use correlated only with play, remediation, enrichment, or reward tends to remain tangential to the learning process and does not necessarily affect all students (Moersch, 2001). Effective technology should be seamless and transparent. It must be naturally intertwined into daily activity much in the same way technology is embedded into daily work or personal tasks for working adults. Truly integrated technology moves beyond teacher-only use and allows students to assume user roles. Lesh, Post, & Behr (1987) argued that mathematical concepts reside not in the physical components of the environment, nor within the prescribed activities, but rather, they reside in student actions and experiences within the environment.

Computer Algebra Systems

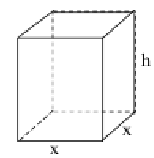
Computer algebra systems are a relatively recent development in the secondary mathematics classroom but have previously radically transformed the teaching of

mathematics at the university level particularly in the area of calculus (Crowe & Zand, 2001; Fey, et al., 2003; Heid, 1984, 1988; Heid, et al., 2002; Heid & Edwards, 2001).

A computer algebra system (CAS), such as Maple or Mathematica, is a software tool that can perform cumbersome numerical computations as well as complex symbolic manipulations, like factoring and simplifying algebraic expressions, factoring polynomials, finding the solutions to a system of equation, and various other manipulations. In calculus, they can be used to find limits, symbolically integrate, and differentiate arbitrary equations. Additionally, CAS often include facilities for graphing equations and provide programming language constructs for the user to define his/her own functions and procedures. An example is shown in Figure 2.

In this example, the student algebraically solves a straightforward optimization problem typical of an introductory calculus course. Note how much of the by-hand computation is performed by the CAS thus freeing the student to think in higher-level terms- terms related to the calculus concepts rather than algebra concepts. Also, notice how the student can add commentary to the document each step. CAS systems also provide extensive graphing capabilities that would also permit solving this problem using a graphical approach rather than the algebraic approach shown.

PROBLEM: A manufacturer wants to design an open box having a square base and a surface area of 108 square inches. What dimensions will produce a box with the maximum volume?



The volume is given by expression

```
> V := x*x*h;
```

$$V := x^2\,h$$

The surface area is given by the expression

```
> S := x^2+4*x*h;
```

$$S := x^2 + 4\,x\,h$$

Because V is to be maximized we would like to expression V in terms of a single variable. Presently, it depends upon two variables x and h To so this we will solve the secondary equation, S=108, and use it to eliminate the variable h from the expression for V.

```
> h := solve( S=108,h); # Maple solves the equation algebraically!
```

$$h := -\frac{x^2 - 108}{4\,x}$$

Now when we examine the expression for V we see that the variable h no longer appears.

```
> V;                    # the new h is automatically substituted
```

$$-\frac{x\,(x^2 - 108)}{4}$$

Now we will use the first derivative test to determine the critical points for V. The extreme values of V must occur at endpoints or critical points. Critical point occurs where the first derivative is 0 or undefined.

```
> critical := solve(diff(V,x)=0,x);
```

$$critical := -6, 6$$

The only critical value with significance is the strictly positive one.

```
> eval( V, x=critical[2]);   # evaluate the Volume at the 2nd point.
```

$$108$$

Now try the endpoints, x=0, x= $\sqrt{108}$

```
> eval( V, x=0);
```

$$0$$

```
> eval( V, x=sqrt(108) );
```

$$0$$

```
> eval(h,x=6);
```

$$3$$

Thus the maximum volume will be 108 cubic units and will occur when x=6, and h=3.

Figure 2. A Sample Session in the Maple Computer Algebra System

This ability to manipulate expressions symbolically created concern over how they will affect student's ability to perform mathematical manipulations and procedures themselves.

> We have much to learn about CAS in the teaching and learning of mathematics. We need to understand how CAS and paper-and-pencil procedures can and should co-exist. We need to learn what paper-and-pencil activities are necessary to use CAS effectively. We need to determine what mental computations in algebra and calculus are important for student learning…We must learn how to teach to the next level above 'doing' or practicing procedures- thinking about mathematics (Waits, Demana, & Kutzler, 1997)

Increasing use of these systems forced important reflection upon instructional design and implementation particularly in the area of introductory algebra and calculus. In particular, (a) Do students who use CAS systems perform as well (or better) than students who do not utilize CAS systems? (b) Are symbolic manipulation tools, such as CAS, a detriment to student's by-hand computational skills? (c) In what ways do CAS systems alter the emphasis on mathematical skills versus conceptual understanding?

With regard to CAS, much of the research has focused upon whether the use of such systems negatively impacts the students ability perform symbolic manipulations by hand. In this review, studies relating the use of CAS in calculus will be emphasized although there is a wealth of studies relating to the use of CAS in algebra courses.

Heid (1984, 1988) utilized a CAS in a college calculus course in which the traditional manual symbolic methods were not taught until near the end of the course.

During the first 12 weeks, two classes of college students (n=39) studied calculus concepts using a computer algebra system to perform routine manipulations. The remaining 3 weeks of the course were spent on manual skill development. Analyzing class transcripts, student interviews, field notes, and test results, Heid found students did as well as those who had not utilized a CAS and additionally that students who used a CAS had better conceptual understandings. Students showed better understanding of course concepts and performed nearly as well on a final exam comprised of routine skills as a class of 100 students who had practiced the manual skills for the entire 15 weeks.

Palmiter (1986) studied 120 students using a CAS in a second-semester introductory calculus course studying integration. These 120 students were assigned to one of three groups, two controls, and one experimental group. All groups covered the same content and conceptual material but the experimental group utilized the Macsyma CAS to perform integrations; the control groups used paper-and-pencil methods of integration. After five weeks, the experimental group was tested on two types of exam, one conceptual that did not permit the use of the CAS and one computational that did permit CAS use. The two control groups were tested after ten weeks. The experimental group scored higher on both exams suggesting that the content could be taught in less time with greater conceptual understanding.

Schrock (1989) compared a class receiving instruction emphasizing computational skills with an experimental group which stressed conceptual understanding. The Maple CAS was utilized in both classes but it was utilized as a simple demonstration tool with the control groups. Student performance on a midterm examination was utilized to measure conceptual understanding and performance on the

final exam was used to assess computational skill. The experimental group showed greater conceptual understanding with no loss of computational skill.

Cunningham (1991) considered the issue that post-test designs that do not permit students taught using CAS to utilize the tool on the post-test might negatively impact results as students were in an unfamiliar setting unable to utilize to key cognitive tool. A pretest-posttest design was utilized to explore the effects on 53 freshman's calculus achievement of using software capable of symbolic manipulation to reduce hand-generated symbolic manipulation.

A software package was used for classroom demonstrations for both the control and experimental group. The control group relied upon traditional pencil-and-paper methods to perform symbolic manipulation and the experimental group relied extensively on the software to perform computer-generated symbolic manipulation, which included the evaluation of limits, differentiation, indefinite integration, and others.

Two similar posttest instruments were developed to measure manipulative and conceptual calculus achievement, respectively. For the administration of the posttest instruments, the experimental group was divided into two Groups, B1 and B2. Group B1 was administered the manipulative assessment with access to the software and the conceptual assessment without access to the software. For Group B2, this sequence was reversed; Group B2 was administered the manipulative assessment, without access to the software, and the conceptual assessment, with access to the software.

On the manipulative sections of both assessments, whichever half of the treatment group had access to the software achieved a significantly higher mean score than the control group at the .05 level; whichever half did not have access to the software

achieved a mean score similar to that of the control group. On the conceptual assessment, Group B2 achieved a significantly higher mean score than the control group at the .01 level. Group B1 achieved a higher mean score than the control but not significantly higher. This study suggests that the use of the software improved achievement and did not cause damaging effects when access was denied. However, success required instructor use in the classroom in tandem with extensive student use both outside of the classroom and on tests.

Cooley (1995) studied the effects on achievement and conceptual understanding of integrating a computer algebra system into an introductory calculus course to determine whether students in a CAS enhanced calculus course developed a higher level of conceptual understanding of key concepts (limit, derivative, instantaneous rate of change, integral, maximum and minimum, and curve sketching) than students in the traditional calculus course. Two calculus classes were studied. One class was enhanced with a computer component that included laboratories written for the Mathematica computer algebra system. The other class was taught in the traditional manner, without technology. Background data were collected from both classes at the beginning of the semester. Both classes completed a conceptual exam at the end of the semester to measure conceptual understanding of the six calculus concepts; limit, derivative, instantaneous rate of change, integral, maximum and minimum, and curve sketching. Five students from each class were interviewed at the end of the semester and discussed various calculus questions.

The two groups of students were very similar in their background characteristics. The only significant difference was that a larger percentage of students in the technology

group had previously completed a high school calculus course. Therefore, previous completion of a calculus course was used as a covariate to compensate for this difference. Students registered for the sections through normal registration procedures.

The students in the technology group scored significantly higher in three of the six conceptual areas: limit, derivative, and curve sketching. The non-technology group did not score higher in any of the conceptual areas. The overall, total conceptual scores were also significantly higher for the technology group. The technology group also scored significantly higher on the traditional calculus questions demonstrating that these students did not suffer any loss of computational skills.

Park and Travers (1996) compared the achievement of 42 students in a traditional non-CAS course with 26 students enrolled in the "Calculus and Mathematica" course. The study utilized concept maps and interviews to assess conceptual understanding. The use of Mathematica permitted students more time to focus upon conceptual aspects of calculus rather than computations. The study concluded that students had a higher conceptual understanding without a significant loss in computational skill.

Although CAS systems have been utilized extensively for relating graphical, visual, and symbolic representations in calculus (Leinbach, Hundhausen, Ostebee, Seneschal, & Small, 1991), No research specifically involving programming in a CAS at the secondary level was found.

Student Understanding of the Limit Concept

The notion of *limit* underlies both differential and integral calculus; it is the key concept underlying such formal notions as continuity, differentiability, and integrability.

Yet a complete understanding of this concept has proven to be notoriously difficult for students to attain (Cornu, 1991; Monaghan, Sun, & Tall, 1994; Tall & Vinner, 1981; S. R. Williams, 1991). Even without a formal introduction to limits students generally possess intuitive preconceived notions of the concept they implicitly bring to bear upon problems they encounter. To understand some of the inherent difficulties consider the following simple examples.

Example 1: What is the limit of the sequence of values 1.9, 1.99, 1.999, 1.9999, ….?

Example 2: What is the limit of $\frac{1}{n}$ as n increases without bound? Mathematically, this is asking the student to evaluate the limit $\lim_{n\to\infty}\left(\frac{1}{n}\right)$.

In response to Example 1, students often give the correct answer of 2 but tend to see the limit as a value that is approached and never reached. In other words, the limit is seen as a *process of becoming something* (becoming 2) rather than that *something* (2). Similarly, in example 2, if students write out the first few terms, 1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9, 1/10,…1/100,…, they get an idea that the values are approaching a small non-negative value and may correctly hypothesize that the limit is 0.

The difficulty enters when they are asked to give proof that these values are in fact the limits. In neither case do any of the terms actually equal 2 (0 respectively), so how is it that the limit is 2 (0 respectively)? In the second example, how do we know that the limit is not just a very small non-zero value, say 1/1000 or 1/100000 rather than 0?

The answer to this puzzle is bound up in the formal definition of sequential limit that educators hope to entice their students to adopt – a definition that is laden with formal mathematical notation and deep conceptual meaning.

Definition (Limit of a Sequence): A sequence $\{a_0, a_1, a_2, ..., a_n, ...\}$ has limit a if, for every positive number $\varepsilon > 0$, there exists an integer N such that whenever $n \geq N$, $|a_n - a| < \varepsilon$.

As will be described, coming to grips with the *limit as process* and *limit as object* duality is at the heart of understanding this definition. Students must ultimately understand that *the limit (object) is known as soon as we define a suitable process for ensnaring the proposed limit*. The very construction of this endless process proves the limit; it entraps the limit object.

In considering these simple examples, it is not difficult to see why many researchers encountered the common misconception that the limit is an abstract *process* that never reaches a concluding value. Supporting and augmenting this finding, Tall & Vinner (1981) asked 70 first-year university students, who had received a grade of A or B in the English A-level mathematics track, to explain the meaning of the specific statement $\lim_{x \to 1}\left(\frac{x^3 - 1}{x - 1}\right) = 3$. Twenty students responded. Of the 17 correct responses, 11 utilized a dynamic description. The researchers then generalized the question by asking students to state a definition of $\lim_{x \to a} f(x) = c$ if they could recall one. Forty-nine students responded. Responses were classified based upon correctness and level of formality. Of the correct responses, 27 were dynamic in form (e.g., as x tends towards the value a, the value of $f(x)$ tends towards c) and four were stated in proper formal mathematical notation. Of the 18 incorrect responses, 14 were attempts using formal mathematical notation. This suggests that process-based notions seem to be the most easily internalized by students.

In responding to these two questions, students generally used the same approach to the respond to the specific limit as to the general definition with one exception. Interestingly, in responding to the question, "explain $\lim_{x \to 1}\left(\frac{x^3 - 1}{x - 1}\right) = 3$," only four of the fourteen students who gave incorrect formal explanations used their formal definitions of $\lim_{x \to a} f(x) = c$ in their approach to the example. The remaining 10 provided a dynamic explanation. For these ten students, the request for a definition apparently evokes a different concept image than is evoked by a specific example. Since they gave incorrect definitions, it appears that their concept definition image may have been weak or faulty. Their failure to use their concept definitions on the example suggests that their concept definition images may be in conflict with, or disconnected from, their concept images of limit.

Heid (1988) reported on two groups of calculus students; one instructed using traditional non-computer based methods and the other utilizing the graphical and symbolic capabilities of a computer algebra system. Both groups came to view the limit as a *process* rather than a number. They tended to focus on the "approaches some number" understanding rather than as the number being approached. This of course ties back to Tall's concept of a proceptual concept.

Williams (1991) identified common notions of limit in ten post-secondary students and attempted to encourage them to adopt a more formal notion. He found the dynamic, procedural notion of limit found in earlier studies was firmly held and attempts toward adopting more formal notions were met with extreme resistance. Students tended

to think in terms of simple functions that supported this dynamic view and considered anomalous cases to be minor exceptions to their dynamic model.

Likewise, (Monaghan, et al., 1994) found the common dynamic process view of limit was much stronger than the limit as an object view. In this study, a group of nine students with prior experience utilizing the *Derive* computer algebra system were contrasted with a group of 19 students with similar backgrounds lacking experience with a CAS. Students completed a questionnaire probing their conceptions of sequential limit from an algebraic and graphical perspective. Students in the CAS groups were permitted to utilize the CAS in responding to the questions. Within two weeks of completing the survey, the students were interviewed. Interestingly they found that the automated processes provided by the CAS could at times obscure and discourage deeper reflection but could simultaneously help promote a more balanced view of limit since the CAS is able to compute a limit as an exact 'proper' entity. This ability to produce an end product rather than a converging sequence of partial sums, as in (Li & Tall, 1993), serves to elicit the object view of limit.

In calculus, students are typically interested in a slightly different notion of limit, the *limit of a function* rather than a *limit of a sequence* as considered previously. This definition is even more formidable as there are now two "limiting" processes occurring simultaneously, a domain process and a range process! Consider the formal definition:

Definition (Functional Limit): Let a and L be real numbers. A function $f(x)$ has limit L as x approaches a, if given *any* positive $\varepsilon > 0$, there exists a $\delta > 0$ such that for all x, $0 < |x - a| < \delta \Rightarrow |f(x) - L| < \varepsilon$. When this is true, we write $\lim_{x \to a} f(x) = L$.

This is a very formal definition that requires a great deal of effort, exploration, and contemplation. It requires understanding the sequential limit with respect to two interrelated limiting processes.

Student Understanding of the Derivative Concept

Conceptual understanding of the derivative has shown to be a similarly elusive goal. Consider its formal definition:

Definition: (Derivative) The derivative of a function $f(x)$ with respect to its input, variable x, is the function $f'(x)$ whose value at x is $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

This very definition involves a limit and, as such, recalls all the related issues just discussed. Further, studies examining the derivative concept (Heid, 1988; Orton, 1983) have documented further difficulties associated with this concept. Not only is it challenging to understand the process yielding the limit but also it is further a challenge to understand and interpret what results.

Orton (1983) studied a group of 110 students - 60 of which were pre-college students. He found that students had more difficulty with questions related to understanding differentiation and graphical approaches to rates of change than with calculating or applying derivatives in specific applications. Students tended to rely on algorithmic steps not requiring conceptual understanding. Heid (1988) found, as with limits, that students viewed the derivative as an approximation to the slope of a line tangent to a graph rather than being the true slope.

Programming and Calculus

There have been a few studies at the college level utilizing programming to teach calculus concepts. Flores (1985) studied two groups of chemistry students enrolled in a first year calculus at a state university. Students spent three hours per week in class and attended one two-hour laboratory a week. Each of the 55 students was randomly assigned to one of two groups. One group wrote programs that developed understanding of the limit and derivative concept; the other group utilized pre-written programs to study the concepts. Prior to the study, both groups attended a four week introduction to programming seminar that did not directly address the limit and derivative concepts. The two groups were contrasted using a two-by-two (treatment by group) analysis of variance on a pre-test and a post-test. The pre-test measured familiarity with BASIC programming commands and syntax and the ability of students to predict the output of a given program. There was no control for prior mathematical knowledge or ability. With a significance level of 0.05, the pre-test results did not indicate significance, $F(1, 52)=1.25$, suggesting that there were not differences in programming ability between the two groups. In both classes, the programmers achieved higher post-test scores than the non-programmers however, neither the main effects, $F(1, 54)=.29$, nor the interaction, $F(1, 54)=0.86$, were statistically significant.

Li and Tall (1993) considered using functions and loops in the BASIC programming language to promote the understanding of the sequential limit. The study was performed at the university level over a period of 20 weeks in a course on programming and numerical methods using a variant of the BASIC programming language. Data was collected using a pre-test/post-test design. The pre/post test contained

questions on limits of sequences. Students were also interviewed and their responses to submitted written assignments reviewed. They noted that a key confounding consequence of finite precision arithmetic was that this approach continued to enforce the very process-based notion they were attempting to surmount. Due to the limits of finite precision, the limit of a sequence was not reached. This served to enforce the view that the limit is a process rather than a single definite number (an object).

Conclusion

Understanding the limit and derivative concepts is a profoundly difficult conceptual task for students and is influenced by a host of different factors. Research salient to the study of programming as a tool in mathematics instruction has been discussed in this chapter. This includes research on programming, the application of computer algebra systems in mathematics, and the conceptual issues underlying student understanding of the limit and derivative. It demonstrates how earlier research into application of programming coupled with a consideration of present use of computer algebra systems provides a unique context for applying programming concepts. The next chapter explores the philosophical orientation and conceptual frameworks guiding the study.

CHAPTER THREE

METHODOLOGY

This chapter describes and justifies the qualitative research methods utilized to investigate the impact of learner constructed programming based visualization tools on understanding the limit and derivative in a secondary Advanced Placement Calculus course. The choice of methodology is guided by the following research questions.

> (1) How does the development of conceptual understanding of limit and derivative contrast between students constructing and utilizing programming based exploratory tools as compared with students utilizing preconstructed exploratory tools in a CAS environment?
>
> (2) What factors influence these two developmental trajectories?

This chapter details the chosen methodology by discussing (a) the specific qualitative methodology adopted, (b) the research environment and participants, (c) my role and bias, (d) the conceptual frameworks supporting the study, (e) the qualitative data collection methods, (f) the methods of data analysis, and (g) trustworthiness.

Qualitative Methodology

Qualitative research provides an in-depth inquiry in words rather than numbers. In such studies, the researcher is not trying to generalize an observation but rather to characterize one particular case in a very rich and deep way- in a way that cannot be captured by a numerical value or values (Creswell, 2003, p. 199). Stake (1995) stated that "In quantitative studies, the research question seeks out a relationship between a small number of variables…In qualitative studies, research questions typically orient to cases or phenomena, seeking patterns of unanticipated relationships." Merriam (1998) stated "Qualitative research assumes that there are multiple realities—that the world is not an objective thing out there but a function of personal interaction and perceptions. It is a highly subjective phenomena in need of interpreting rather than measuring" (p.17).

Within the Constructionist and APOS frameworks, the computer serves as a cognitive tool as elaborated by Pea (1985), Salomon, Perkins, and Globerson (1991), and more recently by Jonassen, Carr, and Yueh (1998). Salomon et. al. suggest cognitive tools are tools with which learner's works in partnership and in which a relationship between learners and tools develops over time. Thus the tool depends upon the learner's cognitive guidance and, in turn, the learner's actions are guided by interactions with the tool. (Jonassen, et al., 1998) defines a closely related concept, *Mindtools*, which are representational tools where the learner is able to externalize their representation and solution to a problem in a way that permits interaction and reflection with the representation.

In this study, students interacted with cognitive tools in two senses. First, they interacted with the Maple CAS which provided the environment for development and

exploration and, second, they interacted with the tools with which they were provided or which they constructed within the CAS environment. Such tools represent the Constructionists' public entities of interaction. Students in the programming class, P, were charged with constructing and subsequently utilizing tools to explore the limit and derivative concepts. Students in the non-programming class, N, utilized functionally similar preconstructed tools to perform identical explorations of limit and derivative. In understanding ways in which the learners forged their understandings, I focused upon *interactions* learners made with their respective tools and each other through the medium of the computer algebra system as well as their interactions within the CAS environment itself. The document framework provided by the Maple CAS provided the ability to record temporally sequenced interactions of learners in the environment as well as with the exploratory tools used or developed within the environment. This record provided an opportunity to more thoroughly explore students' thoughts and actions.

Case-study Methods

To follow these interactions, this study utilized a multiple embedded case design (Yin, 2002) in which two classes of AP Calculus AB explored the limit and derivative concepts of differential calculus. One class, P, comprised a group of students who programmed and used exploratory tools to understand these concepts. The other class comprised a group, N, that did not program the exploratory tools. Instead, they utilized preconstructed exploratory tools to perform identical explorations. These two classes, class P and class N, represent two cases, case P and case N, respectively. Within each of the two cases there were two further units of analysis, collaborative pairs of students and the individual students. Pairs in the P group were denoted $P_1, P_2, \ldots$ and pairs in group N

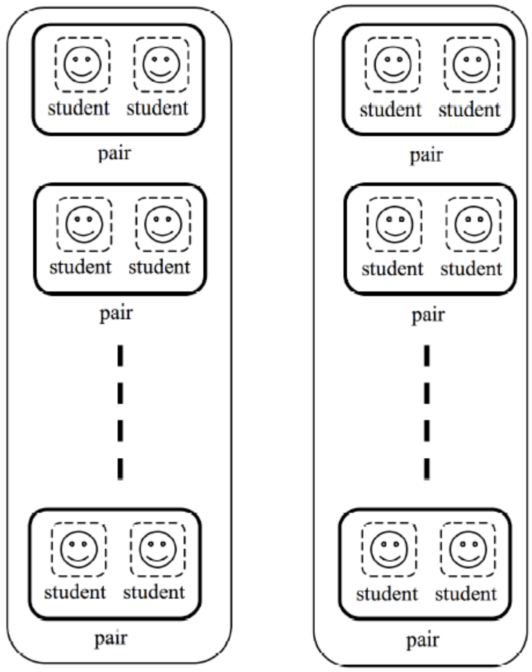


Figure 3. Units of Analysis in the Proposed Multiple Case Embedded Design

as $N_1, N_2, \ldots$ This hierarchical model is shown in Figure 3 in which each unit of analysis is enclosed in a representative rectangular grouping. The study explored the development of conceptual understanding of the two groups, $P = \{P_1, P_2, \ldots\}$ and $N = \{N_1, N_2, \ldots\}$, using a lens of varying magnification as provided by this hierarchical collection of sources.

Stake (1995) described case-study as research studying a program, event, activity, process, or individual bounded by time and activity in which data is collected utilizing a variety of methods over a sustained interval of time. They serve to organize and report upon the actions, perceptions, and beliefs of groups or individuals within specific settings.

In this study, during a time period of one semester, two classes of Advanced Placement Calculus AB students were instructed by myself, utilizing an identical

curriculum, making the research environment bounded by time and activity, what Merriam (1998) refers to as “particularistic” (p. 29). The research focused on the development of conceptual understanding of limit and derivative through an examination of student interactions occurring within a highly specific computing environment.

Data Sources

The developmental steps and processes outlined by the APOS decompositions as well as conceptual understandings and misunderstandings produced were documented at several levels reflecting varying granularities of data in this case-study design. At the student-pair level, students completed assignments in the Maple computer algebra system producing electronic documents, called worksheets, chronicling their explorations in temporal form. These documents provided a significant source of developmental insight as they captured details of the students’ problem-solving processes. Another key data source was research notes taken by myself as students completed assignments. These notes documented the types of questions posed, difficulties encountered, as well as their contexts at all levels of analysis. Finally, at the student-level, students wrote responses to questions on exams and post-lab activities that further elaborated their understandings. These varied sources provided triangulation of work done by the pairs, by individuals, and by the respective groups, P or N, strengthening credibility of the study's findings.

This constellation of data, representing varied magnifications (granularity of data), extends a limited present understanding of the conceptual development of limit and derivative by exploring how specific conceptions and their representations differ, how learners interact with the programs they construct (or use) and how their interactions mediate these understandings.

Research environment

This study was conducted in a secondary school in the southeastern United States with an approximate enrollment of 2015 students in grades 9-12 arrayed in the demographic groups shown in Table 1. The data was taken from the 2005-2006 state report card (the most recent data available at the time of the study). The school was comprised of a full-time faculty of 104 teachers with an average of 15 years of teaching experience.

I am a mathematics and computer science teacher who has taught mathematics for 11 years and computer science for a decade. In this study, I served as a participant-observer by providing classroom instruction outside laboratory periods and by serving as resource for using the computer algebra system and program design while simultaneously observing and documenting student actions and behaviors during lab periods.

Table 1
Demographics of Research Site

Race/Ethnicity	School Percentage	State Percentage
Asian	24	3
Black	16	38
Hispanic	18	8
Native American/Alaskan Native	0	0
White	40	49
Multiracial	2	2

The students utilized the Maple 8 (Maplesoft, 1988) computer algebra systems on a classroom set of laptop computers available on a daily basis. Laboratory periods of 52 minutes initially occurred once a week for 16 weeks. However, as the study progressed it became necessary to have lab periods on two consecutive days a week in order to provide additional continuity for the students. Students had limited access to computers outside formal lab periods before and after school.

Research Participants

Research participants were consenting students enrolled in two in-tact classes of an Advanced Placement Calculus AB. Student pairs in class P programmed and utilized programming-based exploratory tools and student pairs in class N utilized functionally identical, but preconstructed, exploratory tools in Maple to complete lab assignments.

Students were paired so they could utilize a collaborative strategy borrowed from software engineering known as pair-programming. The programming groups of class P naturally implemented this collaborative strategy as they were in the context of software development. However, the non-programming groups of class N also utilized the strategy as they explored using the preconstructed software tools.

The justification for utilizing this model with the non-programming class, N, is based upon findings from an essentially similar strategy of *peer-assisted learning (PAL)*. Peer-assisted learning takes place in an environment in which peers provide active help and support. Topping and Ehly (1998) reported that peer-tutoring leads to more active learning, higher levels of cognitive reasoning, greater transfer, and positive dispositions

toward learning. Additionally, the pair-programming model introduces specific participant roles within the pair that serve to increase the mutual interdependence of the pair on each other. “Students perceive that they can reach their learning goals if and only if other students in the learning also reach their goals.” (D. Johnson & Johnson, 2004, p. 786)

Prior to inclusion in the study, student and parental consent was obtained using the following protocol. After explaining the study in detail to every enrolled student, each was given a consent form and asked to consider participating in the study. Rather than returning consent forms to me, students were instructed to return the forms to a colleague who was responsible for creating the student pairs.

After three weeks, my colleague provided a list of student pairings in which every student in the class, consenting or not, was paired with another student. Because I was unaware of which students agreed to participate in the study, I operated on the assumption that all students were consenting. At the end of the semester, after final grades were submitted, a list of consenting student pairs was provided to me so that I could begin data analysis. This was done to protect the students from feeling pressure to participate or fearing their participation, or lack of participation, would unfairly impact their grade in the course. Students specifically agreeing to participate in this study comprised the data set utilized for analysis.

There is mixed evidence that same-sex pairings are beneficial (D. Johnson & Johnson, 2004; Werner, Denner, & Bean, 2004; L. Williams & Kessler, 2003). Thus, students were paired into same-sex partners during lab periods to the extent permitted by class composition and student consent. Additionally, several studies indicated, in the

context of technology-supported instruction, the highest educational benefits are derived when heterogeneous achievement groups work with technology (D. Johnson & Johnson, 2004). Hence, each pair consisted of a high-achieving mathematics students and lower achieving mathematics student based upon their grade point average (GPA) in secondary mathematics courses. A higher achieving student was a student whose overall mathematics average in secondary school mathematics was 90 percent or higher. It should be noted that the labels higher achieving and lower achieving should not be taken to mean lower achieving students perform poorly in mathematics. These students are enrolled in an Advanced Placement mathematics course and, as such, have been quite successful in prior mathematics courses.

When a complete pairing within a class was not possible, students were placed within a group as a third member and the pair programming model was adapted so two individuals in the group of three have the role of *navigator* (to be described).

The initial pool of potential participants consisted of 21 students from class P and 20 potential participants from class N. In class P, six consenting students comprised four group pairs, P_1, P_2, P_3 and P_4. Of these four group pairs, only two had both students consenting- one group of females, P_1, and one group of males, P_2. The remaining two students formed groups P_3 and P_4 from which there was only one consenting member, a female and a male, respectively.

Eight consenting students from class N comprised four complete consenting pairs. One of the pairs formed a group of females, N_1 and the remaining three of the groups, $N_2, N_3,$ and N_4, comprised pairs of males. These groups are graphically depicted in Figure 4.

Case P: Programming Groups

Group P_1 consisted of two higher-performing female students one White and one Asian. Both students were 17 years of age in Grade 12. Neither was classified as Gifted. Both students had taken an Algebra 2 course taught by myself two years prior.

On the initial survey, one member indicated she "*knew nothing about computers*." Both students had utilized graphing calculators in prior mathematics classes and saw them primarily as a labor saving device. Neither student had any prior programming experience. The pair saw untapped potential for computing technology in the area of online instruction and in demonstrating mathematical concepts. Both students indicated

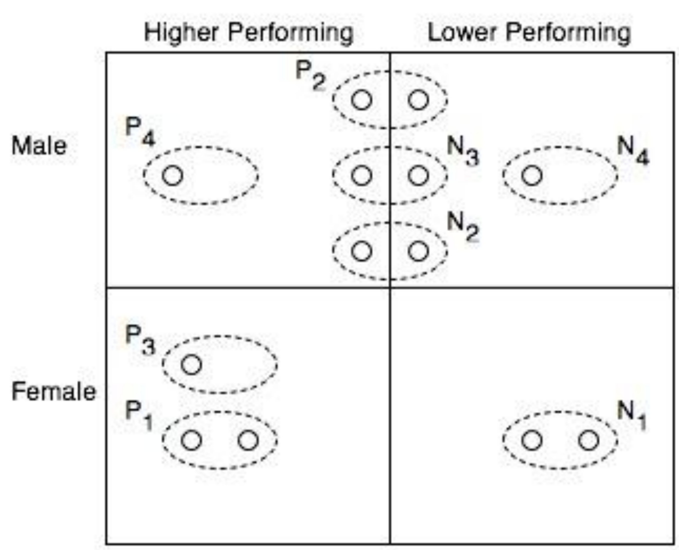


Figure 4. Participant parings based upon gender and prior mathematical performance.

that they had worked collaboratively in prior math course and that such collaboration provided verification of answers, "*to help each with an assignment and check each other's answers and questions*" , and mutual assistance, "*if I didn't understand a problem, my partner could help me and we could work together to get the right answer.* " Their initial conception of limit was identical indicating a dynamical perspective that a limit "*describes how function moves as x moves toward a certain point.*"

Group P_2 consisted of one higher-performing Gifted Asian and one lower-performing White male student. Both were 17 years old in Grade 12 at the time of the study. Neither of these students had prior instructional experience with me. On the initial survey, both students had utilized graphing calculators in prior mathematics classes indicating, as in group P_1, the role of such technology was "*making work faster and somewhat easier*" and "*making calculations easy and allowing more focus on memorizing formulas.*" Neither student had prior programming experience. Both students indicated they worked collaboratively in prior math courses having had group assessments and group work in completing assignments. Initially these two students held differing views of limit. One held a correct formal notion of limit whereas the other held a dynamic view that a limit described "*how a function moves as x moves toward a certain point*."

Group P_3 was comprised of two females only one of which, a higher-performing Gifted White student, consented to participate in the study. She was 17 years of age in Grade 12 at the time of this study. She had not had any prior instructional experience with me. On the initial survey, she indicated prior utilization of graphing calculators to "*to plot points, form matrices*." The perceived role of such technology was "*to check*

work or do problems that cannot be done by hand.” She indicated prior experience working collaboratively to “*check homework assignments we got in groups and compared answers*”. Her initial limit conception was that a limit is “*a number or point past which a function may not go.*”

Group P_4 was comprised of two male students only one of which, a higher-performing student, consented to participate in the study. This student was a 17 year old Asian in Grade 12. He had not had any prior instructional experience with me. The student saw technology as aiding in the construction of “charts and graphs” and indicated an understanding of mathematics is necessary to understanding technology. Evidently, the student does not consider the converse relationship- that technology might aid in the understanding of mathematics. The student’s prior collaborative experience consisted of working with a partner to solve problems and check homework. No prior programming experience was indicated. This student’s initial limit conception was that a limit is “*a number or point past which a function may not go*.”

Case N: Non-programming Groups

Group N_1 consisted of two lower-performing females. One was an 18 years old Asian and the other a 17 year old Asian in Grade 12 at the time of the study. The 18 year old had previously completed an Algebra 2 course with me two years prior and was concurrently enrolled in an Advanced Placement Computer Science course with myself instructing. The other student had no prior instructional experience with me. On the initial survey, these students both indicated they had used calculators to produce graphs in prior courses. They perceived technology as primarily a labor saving device. Both had worked collaboratively to check homework in prior courses. The 17 year old student had

no prior programming experience but the other was concurrently taking a computer science course with me and therefore had extensive exposure to programming concepts. This student, however, struggled in the programming course due in large part to a lack of motivation; this was reflected in much of the work produced within the context of this study. This 18 year old student held that a limit "*describes how a function moves as x moves toward a certain point*," whereas her partner held that a limit is "*a number or point the function gets close to but never reaches.*"

Group N_2 consisted of two 17 year old White male students. One student was classified as Gifted. Neither student had prior instructional experience with me. Both indicated they used graphing calculators in the past but that the primary benefit to using them in mathematics classes was a reduction in labor and error. Both indicated collaboration checking homework and in completing homework assignments outside of classes. The Gifted student indicated he had performed some BASIC programming on his graphing calculator and also some HTML writing in designing web pages. This pair had been close friends throughout their school experience. One held that a limit is "*a number or point past which a function cannot go.*" Whereas the other held that a limit described, "*how a function moves as x moves toward a certain point.*"

Group N_3 consisted of two White male students who were 17 years of age in Grade 12 at the time of this study. The lower-performing member of the pair was classified as Gifted and had taken an Algebra 2 course with me as the instructor two years prior but had no prior programming experience. The other higher-performing student had successfully completed Advanced Placement Computer Science with me as the instructor the prior year. On the initial survey, both students indicated technology was an effective

demonstration tool in the classroom rapidly producing graphs and three-dimensional visualizations. One student previously worked collaboratively on a science project as well as on a geometry project; the other had participated in a study group in a prior mathematics course. Both were comfortable with me due to their prior classroom experiences with me. In this group both students initially held that a limit describes "*how a function moves as x moves toward a certain point.*"

Group N_4 consisted of two White male students. Both students were 17 years of age in Grade 12 at the time of this study. One of these students had previously taken an Algebra 2 course with me as the instructor two years earlier. As with other groups, this pair had used graphing calculators for graphing and perceived the primary role of technology as a labor saving tool. Neither student indicated any prior programming experience. One student held that a limit was a "*number or point past which a function cannot go.*" His partner held a limit described "*how a function moves as x moves toward a certain point.*"

Role of the Researcher

The researcher's role is necessarily complex involving identification of a meaningful topic, formulation of an appropriate research question, and the development a comprehensive research plan. The researcher is further challenged to account for and reduce any personal biases that they bring to the research (Creswell, 2003). The researcher does not want to influence participants in such a way as to force responses that they believe a given person "should" provide. More specifically, in this study, I had the dual responsibility of being a resource to students as they developed programming skill

and explored the specific mathematical concepts as well as being an objective observer. Such a role is commonly referred to as a participant-observer.

Participant-observation provides a unique opportunity to gather data. Since the researcher is also a participant in the study, they often have access to events they might otherwise not have access. The researcher often develops higher levels of trust with the participants who are more at ease and willing to confide their true feelings and ideas without the fear of disclosure or other consequence (Yin, 2002).

One major problem related to being a participant-observer has to do with the introduction of the researchers personal bias (Yin, 2002). To begin to develop and awareness of my personal biases and to make them as explicit as possible, I include a discussion of my personal experiences and beliefs as they relate to this study. The first step towards understanding the impact of bias in a study is enumerating what biases might exist.

This research has its genesis in my own educational experience both as a teacher and a student. I am an electrical engineer turned educator who has been teaching mathematics and computer science at the secondary level for the past decade. My interest in teaching this combination of content stems primarily from two key experiences: an exposure to computer programming upon entering high school and my undergraduate experience as a math major.

I began learning to program at the age of 13, about the time that I began my first year of algebra. In algebra, I developed a fascination for the expressiveness of formal mathematical notation and, in programming, the ability to outline a process to a machine. These two ideas seemed to harmonize with one another. Such formal notation could not

only concisely describe a mathematical concept but could also describe a process for producing an answer; although I did not know it then, this was my first brush with process-object duality.

As I came to see programming as describing process, I recalled the mathematical processes I already knew: addition, subtraction, and division. Long division had always troubled me. I recall learning long division and being mystified as to how the steps actually produced the correct answer; I had no idea how the process I had been taught worked. I could apply the process and get the correct answer every time yet was totally unaware of what was happening conceptually. Learning to program gave me the tools for developing an understanding of processes like long division; it provided conceptual tools for analyzing and understanding such processes. As I studied new mathematical ideas in school, I would continue to find application and deeper understanding of these concepts through programming. Thus, I have a clear bias that programming can be an empowering tool that can powerfully influence mathematical thinking and learning.

In college, I also studied electrical engineering in which I continued to find application of programming. I became much more aware of the vast representational possibilities for programming- Programming could be used to process human speech, process video, describe the physical layout of electrical circuits and simulate their operation. This served to convince me of the immense value of process-object understandings in mathematics.

A second incident that profoundly influenced my views of mathematical thinking, learning, and understanding occurred in my sophomore year of college. I had a very meaningful discussion with one of my mathematics professors relating to the

development and construction of mathematical proof. In the discussion, it was suggested that to produce a proof, a fruitful first step is to produce some concrete examples that suggest and exemplify ones tentative conjecture. This would be followed by exploration and generalization of these examples. In the course of these explorations, a formal proof would be constructed in non-specific, yet mathematically rigorous, terms not tied to any specific example. Upon completion of the proof, all the examples were to be discarded and what remained was a mathematical proof.

This struck me as bizarre as it seemed to me that the most interesting and illuminating aspects of the proof had to do with the motivations for the steps in the argument. What motivated the conceptual leaps in the argument? Almost certainly they were motivated by their explorations of the examples they had just thrown away. As a math major, this made it very clear to me why I (and assumedly many other math majors) found the canonical *theorem…proof…theorem…proof* format found in most college texts extremely daunting. As I read a proof, there were no motivating examples suggesting why particular steps were taken. Why did they decide to take this step? What made them think that this step would follow from the previous step? Why did they decide to ever pursue a proof?

I found that for me, the best way to understand and decipher a mathematical proof was to try to recreate motivating examples as I read the proof. What I sought was a historical perspective from which to reflect. It was equally intriguing how this reverse engineering task was essentially an exercise in fictional writing. The examples I created were, in all likelihood, very different from those of the proofs author. This fictional history didn't need to be the author's "truth" in order to have immense descriptive and

conceptual power. The possibility of such alternative realities and the need to understand them further suggested the use of qualitative research methods (Merriam, 1998, p. 17).

Thus, I believe there is great necessity for contextual understanding both in learning mathematics and in assessing student understanding. The proposed research design attempts to capture some of that historical context by documenting student actions in the computing environment- to try to recover some of their motivating thoughts and ideas.

Two years into my teaching career, I was asked if I had an interest in teaching Advanced Placement Computer Science. Given my prior background in programming, I naturally welcomed the opportunity. While not specifically a mathematics class, I naturally tended to expose my students to mathematical problems among others. As I observed students, it seemed that, in much the same way programming had affected me, programming helped students develop key mathematical ideas. Interestingly, as these mathematical understandings developed, these key mathematical ideas were often seen as peripheral to the task at hand by the students- the job was to finish the program and produce a working program, the mathematics required was simply a necessary step in attaining the greater goal.

For example, one such assignment was to write a program to add two fractions and give the answer in simplified form. This project, while simple enough to perform in one’s head, is conceptually formidable. Students had to, not only, recall their knowledge of working with fractions but they had to express the knowledge in a systematic and organized way that resulted in a process producing the required result. The project involves notions of divisibility such as common denominator and greatest common

divisor. In implementing processes for each of these concepts, students truly reflected and deepened their understanding of these mathematical concepts. Again, I see the use of programming as a powerful tool for understanding and learning mathematics.

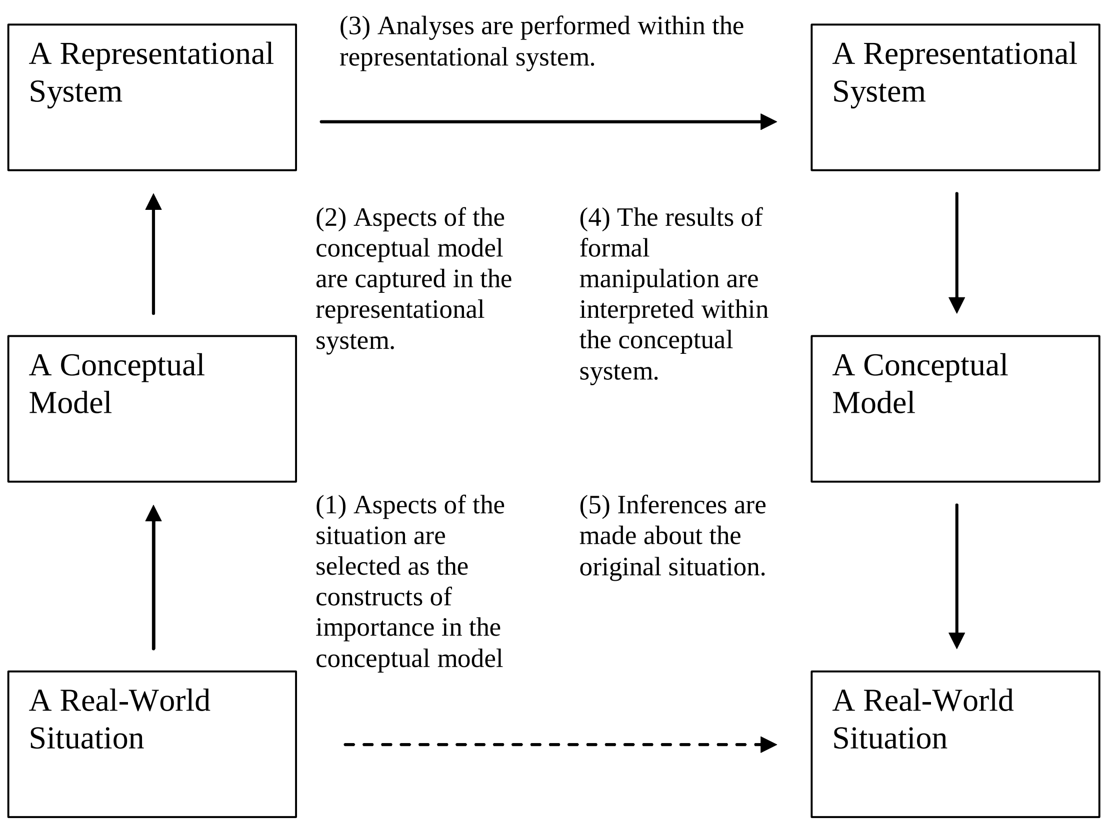
In this research, I wanted to better understand the ways in which programming promotes, or fails to promote, deeper mathematical understanding in the context of learning two key concepts in differential calculus. It is folly to assume that programming is always effective. What aspects of mathematical learning do programming support? Which are not? Why? This is actually a very broad set of questions that certainly cannot be addressed by a single study. Thus, this study sought to explore programming's efficacy in one very specific context.

Role of the Conceptual Model

Schoenfeld (2002) states, "whether or not researchers believe that they have theoretical perspectives and biases, they do. (Researchers who think otherwise are like proverbial fish what are unaware of the medium in which they swim.)" These inherent frames provide lenses through which to see. The phenomenon that we wish to observe will affect our choice of method, which will in turn constrain what aspects of the phenomena we are likely to see or are capable of seeing. Thus, it is important to understand what frameworks support the stated research question.

The research design models a complex real-life situation influenced by many factors, some explicitly known, others not. Thus, the model necessarily excludes aspects of the real situation in order to make study tractable. Such is the genesis of the *conceptual model* of the real situation (see Links 1 and 2 in Figure 5).

Once a conceptual model is in hand, study is undertaken within the adopted conceptualized model (see Link 3). Any conclusions must be understood *in the context of this conceptual model*. Thus, significant findings are not necessarily significant in all contexts; for example, perhaps a neglected aspect of the real-life situation negatively interacts with aspects included in the model, i.e. the effects are mediated by a factor not included in the conceptual model.



Schoenfeld (2002), p. 450

Figure 5. The Role of a Conceptual Model

The development of the conceptual model requires selection of relevant aspects of the real-real world situation that are to be included and excluded. This occurs along Path 1 of Figure 5. These choices are guided by a researcher's theoretical perspective.

Consider the conceptual model of this study, shown in Figure 6. In this model, I have specific target mathematical concepts for pairs to explore and understand (figuratively depicted in the scroll of Figure 6). Using the theoretical framework of Constructionism and APOS theory, I designed and constructed exploratory mathematical visualization tools for pairs to use in the exploration of these two concepts. The specific APOS decompositions for understanding the limit and derivative are shown below in Figure 7 and Figure 8. Students in the programming groups had the supplementary task of implementing these designs prior to their exploratory utilization. All constructions and

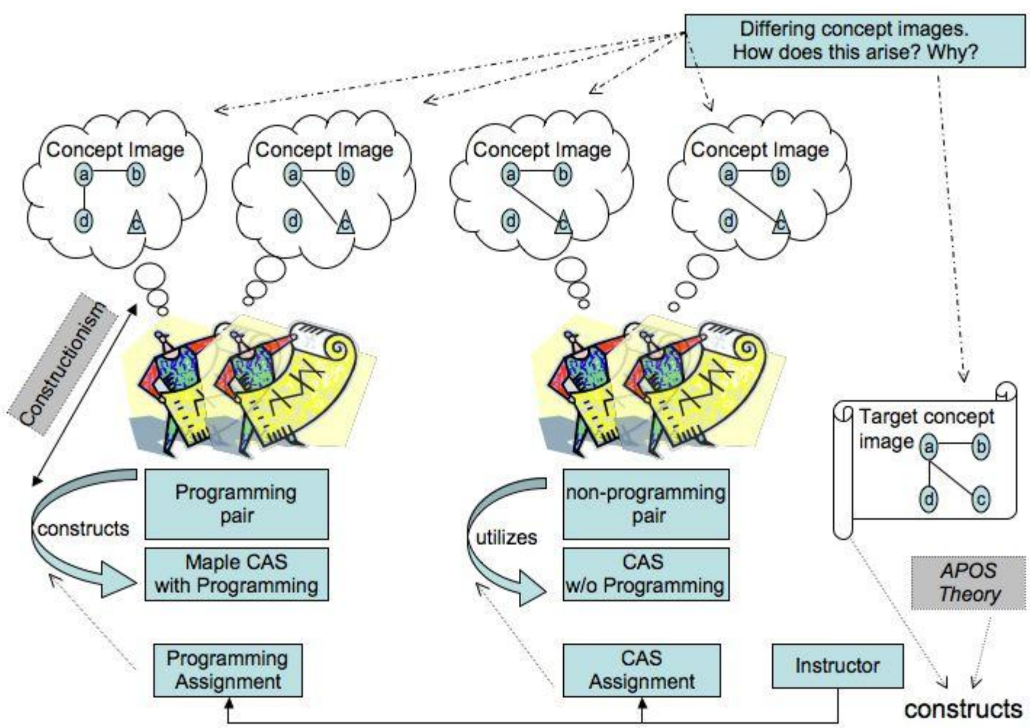


Figure 6. The Conceptual Model

explorations all took place within the Maple CAS environment.

In implementing and utilizing these tools, each student developed his/her own unique conceptual understanding of the target concept that did or did not coincide with the intended target concept. This is depicted in Figure 6 by concept images that differ in appearance from the target concept. The resulting unique conceptual understanding is what Tall and Vinner (1981) refer to as a student's *concept image*. It refers to everything in a student's mental structure associated with a particular concept, which includes mental pictures, associated properties, and processes. Such understanding is formed over years of experiences of many kinds and changes as the individual meets new stimuli and matures (Tall & Vinner, 1981). It is the nature of these developmental similarities and differences that this research sought to expose and understand.

More concretely, consider understanding the limit concept discussed in chapter two. Understanding the limit entails understanding the limit as a proceptual concept, as both a process and an object. This dual natured concept was the desired target concept that I desired students to understand. In prior studies of limits, it was observed that a process-based understanding of the limit was quite common among students. In fact, these studies showed it is exceedingly difficult to foster a different view (S. R. Williams, 1991). This research sought to expose the similarities and differences in conceptual understanding and its development by considering the ways the individual concept images agree with or differ from the intended target concept. Specifically, (a) how does the development of conceptual understanding of limit and derivative contrast between students constructing and utilizing programming based exploratory tools as compared

with students utilizing preconstructed exploratory tools in a CAS environment? and

(b) What factors influence these two developmental trajectories?

Key Assumptions in Conceptual Model

Primary to the conceptual model were two underlying assumptions:

1. Understanding the *limit* and *derivative* concepts can be fostered by scaffolding specific pre-conceptual understanding as outlined in Figure 7 and Figure 8 and,
2. Those conceptions may be scaffolded by *constructive* means and interactions within the Constructionist framework.

APOS Decompositions

The following two decompositions derived in two previous APOS studies (Asiala, et al., 1997; Cotrill, et al., 1996) were utilized to develop programming and non-programming laboratory activities for students to perform. APOS decompositions suggest a specific sequence of objectives and instructional goals to achieve prior to attaining a complete conceptual understanding. As depicted in the conceptual model of Figure 7, I utilized these decompositions to design activities fostering the development of the specific target concepts outlined. In addition to providing a framework for the development of exploratory tools, the APOS theoretical framework also informs the assessment of specific developmental milestones.

1. The action of evaluating a function $f(x)$ at a single point x that is considered to be close to or even equal to, a.
2. The action of evaluating the function $f(x)$ at a few points, each successive point closer to a than was the previous point.
3. Construction of a coordinated scheme as follows.
    a. Interiorization of the action of step 2 to construct a domain process in which x approaches a.
    b. Construction of a range process in which y approaches L.
    c. Coordination of (a) and (b) via $f(x)$. That is the function $f(x)$ is applied to the process of x approaching a to obtain the process of $f(x)$ approaching L.
4. Perform actions upon the limit concept by talking about, for example, limits of combinations of functions. In this way Step 3 is encapsulated to become an object.
5. Reconstruct the process of Step 3(c) in terms of intervals and inequalities. This is done by introducing numerical estimates of the closeness of approach, in symbols $0 < |x - a| < \delta$ and $|f(x) - L| < \varepsilon$.
6. Apply a quantification schema to connect the reconstructed process on the previous step to obtain the formal definition of the limit.
7. A completed $\varepsilon - \delta$ conception applied to specific situations.

Note: In this decomposition, there exists a transition from a procedural understanding in steps 1-3, to a conceptual understanding in steps 4-7.

Figure 7. APOS Decomposition for Understanding the Limit (Cotrill, et al., 1996)

1) **Pre-requisite Knowledge:**

Graphical representations of mathematical objects:	*Coordinating representations of points with a function:*
a) Graphical representations of a point. b) Graphical representations of a line including the concept of slope.	a) Graphical interpretation of (x,y) when y is given by $y = f(x)$. An action conception is indicated when a student has a need for a formula for the function. b) Overcoming the need to have a formula for the function. In the graphical situation, the process of interpreting a point on the graph.

2) **Pathways to the Derivative:**

***Graphical Path to the Derivative**:*	***Analytic Path to the Derivative**:*
a) The action of connecting two points on a curve to form a chord (a portion of the secant line) through the points together with the action of computing slope of the secant line.	a) The action of computing the average rate of change by computing the difference quotient at a point.
b) Interiorization of the actions in (a) to a single process as the two points get "closer and closer" together.	b) Interiorization of the actions in (a) to a single process as the difference in time intervals get "smaller and smaller", i.e. as the length of the time intervals get closer to 0.
c) Encapsulation of the process (b) to produce a tangent line as the limiting position of secant lines and also produce the slope of the tangent line at a point on the graph of a function	c) Encapsulation of the process in (b) to produce the instantaneous rate of change of one variable with respect to another.
d) Interiorization of the processes in steps (a) and (b) in general, to produce the definition of the derivative of a function at a point as the limit of a difference quotient.	d) Interiorization of the processes in steps (a) and (b) in general, to produce the definition of the derivative of a function at a point as the limit of a difference quotient.

3) **Graphical Interpretation of the Derivative:**	**4) Using the Concept of the Derivative:**
a) Graphical interpretation of a derivative at a point. 1. Overcoming the need to differentiate some formula. 2. Coordinate with (1) to see $f'(a)$ as the slope of the tangent line. 3. Coordinate several interpretations of $f'(a)$. The student brings together the ideas of limit of difference quotient, average velocity, marginal cost, etc. and is able to move between interpretations. b) Graphical interpretations of the derivative as a function 1. Seeing the derivative as the function, $x \rightarrow \text{slope at } (x, f(x))$. 2. Identifying $f'(a)$ with the tangent line at a point.	Several coordinations to get the graph of $f(x)$ a) Graphical interpretation of $f(x)$, for a single x. b) Interpretation of $f'(x)$ for a single x as the slope. c) Process of x moving through an interval i. Monotonicity of the function and sign of the derivative ii. Infinite slope (vertical tangent) and infinite derivative iii. Concavity of the function and sign of the second derivative d) Drawing a complete or fully representative graph.

Figure 8. APOS Decomposition for Understanding the Derivative (Asiala, et al., 1997)

Data Collection Procedures and Instrumentation

Within the framework of this conceptual model, I consider data collection as outlined in this model. The primary sources of data were written and electronic documents produced by group pairs, classroom observations, written responses to post-lab activities, and a reflexive research journal (see Figure 9).

Procedures

At the start of the semester, prior to any activities, all students responded to a brief initial informational survey. The purpose of this survey was to develop a context of prior mathematical experience, prior programming experience, and the perceived role of computers in the learning of mathematics. The results of these surveys were discussed previously with the introduction of the participants. A copy of this survey appears in Appendix A.

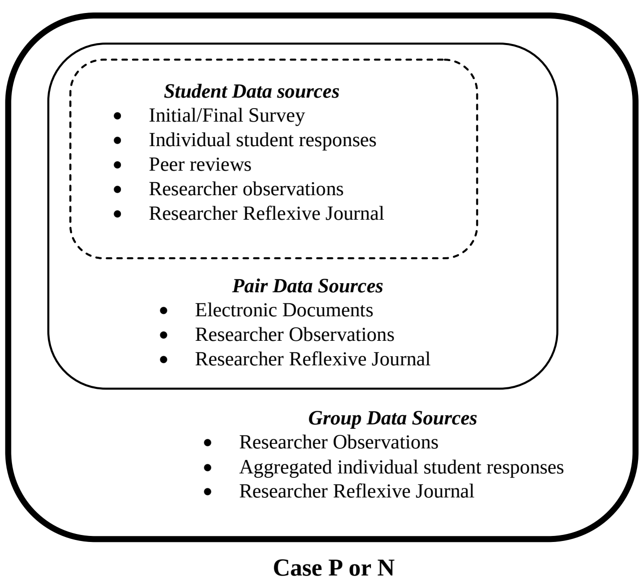


Figure 9. Overview of Data Sources

A coin-flip determined which of two in-tact classes comprised Case P and which comprised Case N. Students worked in pairs only during designated lab periods. If a partner was absent during a given lab period, the participant present was permitted to proceed with the lab activity individually. However, they were asked to summarize the lab and work performed to their peer upon their return. Once paired, students retained their groupings for the duration of the semester. Students received identical classroom instruction outside the lab setting.

Students in all groups explored the mathematical concepts in pairs modeled on the pair-programming software development methodology (Beck & Andres, 2004). In pair-programming, each group member assumes a role, a *driver* who interacts directly with the computing environment and a *navigator* who proposes corrections, and oversees the events taking place. The pair continually interacts with frequent communication between the pair. Periodically, the individuals interchange roles as well; for example, the pairs may alternate driver and navigator roles upon completing the solution to one problem prior to beginning the next. Thus, rather than establishing a peer-tutor relationship, in which one student tutors the other, or a cooperative relationship, in which the work is divided into disjoint and independent parts, pair-programming establishes *collaborative* arrangement consistent with constructivist learning principles.

Research into pair-programming has suggested pair programming yields software with fewer bugs and promotes more interaction among software developers at the expense of additional development time as debugging is typically more costly than initial development (L. Williams, Wiebe, Yang, Ferzli, & Miller, 2002). In this study, however,

the importance of the approach relates to the forms of collaborative interaction it promotes.

Participant pairs were heterogeneous with respect to prior mathematics achievement (GPA in mathematics classes taken at the secondary level) and homogeneous with respect to gender. A higher achieving student was paired with a lower achieving student of the same gender based upon their prior secondary mathematics grades to the extent possible. With regard to gender considerations, there is mixed evidence concerning the impact of technology-assisted instruction on males and females. In the realm of pair programming within the software development industry, there is little support for gender-based pairing (L. Williams & Kessler, 2003). However, at the middle school level some research has suggested that same-sex pairings are more beneficial for females (Werner, et al., 2004).

While awaiting student consent and pairings, students were familiarized as a class with the Maple computer algebra system and with pair-programming as a collaborative strategy (Werner, et al., 2004; L. Williams & Kessler, 2003). Since both of these were new to the students, an initial lab period with ad-hoc pairings served to illustrate how the lab activities would proceed, the challenges, responsibilities, and benefits of pair-programming, and what is expected of each team member.

Once pairs were provided, one day each week (but later two) students in both groups completed corresponding labs in the Maple CAS related to the limit or derivative concept. Each pair worked toward understanding the same mathematical concepts within the APOS framework however, case P labs developed familiarity with necessary programming constructs and programmed the necessary visualization tools whereas case

N labs used the CAS systems as more commonly utilized, as a visualization tool, using pre-constructed exploratory tools. The groups of Case N did not learn programming related content nor were they asked to construct visualization tools but rather used pre-constructed tools reinforcing the specific mathematical objectives outlined in the respective APOS decomposition. The mathematical objectives were identical for both Case P and Case N. Copies of the laboratory activities can be found in Appendix B.

During the lab, the students alternated roles of driver and navigator. The Maple document provided a temporal record of development and/or use of the exploratory tools. Since labs had specific instructional goals correlated with specific themes within the APOS decompositions, a great deal of relevant learning interaction was chronicled within these documents.

As pairs completed labs, I functioned as a participant-observer circulating around the lab observing and documenting interactions taking place between pairs and responding to Maple and/or programming related questions. As different physical classes represented each of the two cases, I was better able to focus on the specific needs of each case.

Upon completing the lab, the pair saved their Maple document and completed a peer review of their partner. Students received a participation grade based upon completion of the lab and the partner's peer-evaluation. The peer-evaluation instrument was due to Williams et al. (2002) who used the instrument in an introductory computer science course at the post-secondary level. A copy of this form appears in Appendix C. Each lab assignment was worth 100 points, 70 points were earned for completing the lab, 20 points were earned for completing a peer review of their partner, and the remaining 10

points were earned based upon the peer review. For example, if their peer review had a score of 80/100, then the student earned eight of the 10 possible peer review points for a total score of 70%+20%+8%=98%. Thus the lab grade was based upon completion and peer review. Each individual student's lab grade is computed by multiplying the peer-evaluation score by the grade on the lab. Williams et. al. (2002) found a similar strategy compelled students to do their fair share of the work.

Routine identical examinations were given to both classes; both groups took the same exams. Student's written responses to questions posed on assessments and post-lab activities correlated to laboratory activities, were examined for evidence of conceptual growth or difficulties. At the end of the semester, a final conception of limit and derivative was assessed on the final exam.

Table 2 enumerates the types of data collected, when gathered, and how it proved useful. The remainder of this section provides further elaboration on data gathering.

Table 2
Data Collection Summary

When?	What?	How?	Who?	Why?
Beginning of semester	Informational Survey	Written Response	All Students (Individual)	Provided context of prior mathematical experience, demographical information, and attitudes related to the role of computers in mathematics learning
Bi-weekly, during lab period	Classroom Observation	Written reflective journal	Instructor	Captured the overall learning environment and documented peer interactions and instructor interactions.
Bi-weekly, during lab period	Maple Labs	Students archive completed lab from prior week	All Student Pairs	Electronic documents provided a temporal log of student activity outlining their solution processes.
At completion of each lab	Peer evaluation	Written evaluation of peer performance	All Students	Provided additional motivation for students to be active contributors to collaborative teams. Also provided feedback on issues related to pairings
Days of Assessment	Written Responses to Problems	Written Response	All Students (Individual)	Probed further thoughts about lab activities and written work
End of Semester	Final Exam – Free Response	Students respond in writing.	All Students (Individual)	Provided opportunity for students to state in writing their finial limit conception.

Document Artifacts

The use of written documents as a method for collecting data in a case study is well regarded (Merriam, 1998; Stake, 1995; Yin, 2002). Yin writes, “Except for preliterate societies, documentary evidence is likely to be relevant to every case study topic.” (Yin, 2002, p. 85) Written documents are stable and unobtrusive observational data, and “quite often, documents serve as substitutes for records of activity that the researcher could not observe directly” (Stake, 1995, p. 68).

In this research, Maple laboratory experiences were electronically recorded in Maple worksheets. Also, students provided feedback on their partner's engagement in the laboratory exercises. Every student completed these evaluations upon completion of each lab activity. These evaluations provided additional documentation relating to student participation and level of engagement in the laboratories. They also provided additional information relating to the perceived effectiveness of the pair-programming model of interaction. They further provided a gauge of the effectiveness of current pairings and permitted alterations if social issues were limiting the efficacy of the lab experiences.

All students took routine assessments in the form of tests and quizzes. Questions directly correlated with objectives in the APOS decompositions as well as common limit conceptions were posed. These additional samples of individual student further permitted exploration of the individual's conceptual understanding, and provided opportunities for students to demonstrate the depth of their understanding.

The final exam consists of two parts, a multiple-choice exam and a written component. The written component will provide an additional work sample for review at the end of the semester. This exam will provide the opportunity for individuals to state, in formal mathematical terms, their formal conceptions of limit and derivative.

Observations

Observations are a common data source in case studies (Merriam, 1998; Stake, 1995). Observations are field notes taken by the researcher on the behaviors and activities of individuals at the research site. In this study, these notes served to contextualize the environment and document interactions and lines of inquiry within the groups during labs. These observations documented what partners said to one another in their groups, what they shared with the class as a whole, and what they shared with me as they engaged in activities and discussion.

As a participant observer, I had the dual role of being the instructor for the course addressing course content as well as being an observer recording classroom interactions. Thus, I was openly identifiable and both cases P and N were aware of their being observed.

Reflexive Researcher Journal

A reflexive researcher's journal was kept chronicling both events in the classroom as well as in the laboratory. The journal will record activities, ideas and decisions made during the research timeline. The journal will serve to provide a calendar of events, document deadlines and identify states of progress. Additionally, the journal served as a personal diary of notes regarding the researcher's perceptions, feelings, and interactions with participants.

Role of Computer Software

To familiarize students with the CAS environment, students were provided with guidelines as to how to journal within the CAS system so that they can write conjectures

and ideas within the electronic documents as they explore concepts. These reflections and historical records of their interactions within the environment will serve as documents for qualitative thematic analysis.

These multiple varied sources of data: (a) observations of the laboratory environment, (b) Maple electronic documents, (c) student responses on assessments, and (d) peer evaluations within a multi-case research design directly address the stated research questions and provide triangulation.

Data Analysis

This research focused upon students' conceptual understanding of two well-defined concepts, the limit and derivative. This development was explored in two different contexts, a programming context, case P, and a non-programming context, case N. The research design can best be considered a multiple case embedded design in which each class represented a case, the highest level of analysis in this study. Conceptual understanding of students in each case was explored utilizing data of varying granularity.

In analyzing the data, both within-case and cross-case analyses were performed. Within-case analysis "means that the researcher identifies themes within a single case" (Creswell, 1998, p. 252), and provides a detailed description. Thus a within case analysis will address interactions related to the APOS decompositions and common limit conceptions taking place within each of the two cases. This entailed the exploration of common trends within the student pairs in each group as well as common trends among the individual students within each pair. Given two laboratory settings, it is worthwhile to consider what was happening with respect to the students in each individual case prior to contrasting across cases (Merriam, 1998).

Whereas, cross-case analysis involves examining themes across cases to discern themes that were common and unique to all cases, it generally follows within-case analysis when multiple cases are studied. Hence, an exploration of the ways in which the various aspects of the two laboratory settings lead to differences in conceptual understanding of students in one setting versus another was addressed via a cross-case analysis.

Case study methods rely on multiple sources of evidence, which converge in a triangulating fashion, and benefit from the prior development of theoretical propositions to guide data collection and analysis (Yin, 2002). The APOS framework, in addition to providing guidelines for structuring laboratory activities, served to inform the data collection process by explicitly stating which mathematical constructions: actions, processes, objects, and schemas are to be sought. The data collected documented specific conceptual milestones suggested by the APOS decompositions. Moreover, they provided a record of various *paths toward* as well as *impediments to* these constructions as they were encountered and addressed by the students.

Document Analysis

Trends in the data were sought utilizing a thematic analysis. Thematic analysis provides a means of encoding and analyzing qualitative information (Boyatzis, 1998). In thematic analysis, qualitative data is first encoded through finding themes. A theme is a pattern in the data that may organize or describe observations or interpret some phenomena. Themes may be derived inductively through a data-driven examination of raw data or they may be derived from prior research or theory. In this study, key themes were suggested by the APOS decompositions themselves and by common notions of

limits found in prior studies; they suggested the kinds of evidence sought in support of the respective learning goals.

This study considered the actions of the pairs within the environment and characterized the patterns of exploration within the document. By studying documents created by students in both laboratory settings, thematic analysis uncovered different patterns of conceptual development and exploration and thus help characterize learning differentials.

Trustworthiness of Results

While quantitative research relies on measures of reliability and validity to evaluate the utility of a study, qualitative research must be evaluated by its "trustworthiness." Coined by Lincoln and Guba (1985), this term is used to represent several constructs including: (a) credibility, (b) transferability, (c) dependability, and (d) confirmability.

Credibility

The credibility of conclusions in a qualitative study is comparable to the concept of internal validity in quantitative research. Lincoln and Guba (1985) suggested that research results be scrutinized according to three basic questions: (a) Do the conclusions make sense? (b) Do the conclusions adequately describe research participants' perspectives? and (c) Do conclusions authentically represent the phenomena under study?

To enhance credibility, this study utilized three common strategies: prolonged engagement, persistent observation, and triangulation (Erlandson, Harris, Skipper, & Allen, 1993; Lincoln & Guba, 1985; Yin, 2002). Prolonged engagement required that I

spend sufficient time in the research context to develop understanding of the events being observed while simultaneously mitigating distortions due to my biases and presence in the research environment. Persistent observation was required if I was to have a temporally consistent understanding of the context. Often, observations will be only relevant when I am able to view them in relation to a specific contextual setting. Finally, triangulation provided a way to perform consistency checks among data sources by collecting and studying information about events and relationships from varying perspectives. According to Lincoln and Guba (Lincoln & Guba, 1985), triangulation is the corroboration of results with alternative sources of data. This permits the researcher to contrast the multiple realities of various observers and seek commonalities and consistencies among them.

In this research, since I am the course instructor there was necessarily prolonged engagement with the students. I saw the students on a daily basis in class and in a laboratory setting weekly. This provided a consistent and timely perspective of classroom context as recorded in my reflexive journal as well as a laboratory perspective chronicled in laboratory observations. Additionally, the collected data represented varying perspectives of the students, the student-pairs, and me. This varying lens provided multiple perspectives from which to triangulate.

Transferability

Similar to the concept of external validity in quantitative studies, transferability seeks to determine if the results relate to other contexts and can be transferred to other contexts (Lincoln & Guba, 1985; Miles & Huberman, 1994). In this study, I sought to enhance transferability by providing a thick, rich description of the contexts,

perspectives, and findings that surrounded participants' experiences. By maintaining a detailed field log of all activities, contacts, and procedures, as well as keeping a current reflective journal of the researcher's experiences, the study provided sufficient description to enhance the transferability of findings.

Reflexive Journal

A reflexive researcher's log chronicled both events in the classroom as well as in the laboratory. The journal recorded activities, ideas and decisions made during the research timeline. The journal served to provide a calendar of events, document deadlines and identify states of progress. Additionally, the journal served as a personal diary of notes regarding my perceptions, feelings, and interactions with participants.

Dependability

Similar to the concept of reliability in quantitative research, dependability refers to whether or not the results of the study are consistent over time and across researchers (Lincoln & Guba, 1985; Miles & Huberman, 1994). Providing a detailed description of how data was collected, and analyzed in the context of the conceptual framework so that another researcher might repeat the study and find similar results enhances dependability. While this study did not provide replication, it did provide broad and deep contextual details that make replication feasible.

Confirmability

Confirmability assumes that the findings are reflective of the participants' perspectives as evidenced in the data, rather than being a reflection of my own perceptions or bias. Confirmability was enhanced by stating explicitly my assumptions

and biases regarding the research (Erlandson, et al., 1993; Lincoln & Guba, 1985; Yin, 2002), having a clear and specific theoretical framework driving the data collection and analysis, and by utilizing multiple varied data sources. Since students had the opportunity to write responses for themselves as individuals, independent voices from each pair provided further triangulation.

Limitations

The goal of this research was to characterize and better understand the conceptual development and understanding of the limit and derivative concepts in two specific CAS contexts- tool developers versus tool users. As discussed previously, in constructing a conceptual framework, there are always aspects of the real-life situation that are excluded. In this study, many aspects not specifically addressed. I, of necessity, made many choices and assumptions as the study was performed. Students were programming in pairs, they were utilizing a specific CAS environment, and they were journaling as they worked within the environment. What aspects of their success and/or struggle were due to their personal interactions with peers? Which were due to usability and operational constraints of the CAS environment? What was the impact of student writing on their learning accomplishment? Certainly, all these impact the effectiveness of the instructional model yet many were not specifically considered within this study.

Summary

This outlines a research agenda utilizing a qualitative methodology to capture the experiences of two groups of students as they developed an understanding of the limit and derivative concepts within a computer algebra system. Based on a review of the

literature, this study addressed a gap in existing studies on the use of programming in mathematics education by proposing a unique application of a computer algebra system to develop and chronicle aspects of mathematical learning and understanding of two key calculus concepts at the secondary level. A qualitative approach utilizing case-study procedures and analysis was utilized. Data from multiple data sources- electronic documents, classroom observations, student written responses, and reflexive researcher notes – were analyzed for conceptual trends using qualitative analyses. The use of varied data sources bolstered validity and reliability. Chapter four presents the results of the data analysis and a discussion of them as they relate to the research question.

CHAPTER FOUR

RESULTS

The purpose of this case study was to investigate, characterize, and contrast the development of student understanding of the limit and derivative concepts from calculus as cultivated using pre-developed tools versus using student-developed (constructionist) programming-based tools. This study took place in a mid-sized suburban public school system located in the southeastern region of the United States of America. Following a typical introductory instructional sequence introducing the limit concept, students were surveyed to gain an understanding of their initial conceptions of limit. Subsequently their development of understanding was chronicled and analyzed using laboratory assignments tailored to the refinement of this initial understanding toward the formal $\varepsilon - \delta$ notion. The examination of this developmental process is directed toward the essential research questions:

(1) How does the development of conceptual understanding of limit and derivative contrast between students constructing and utilizing programming based exploratory tools as compared with students utilizing preconstructed exploratory tools in a CAS environment?

(2) What factors influence these two developmental trajectories?

The findings emerging from the data are shared in two chapters. In this chapter, context is provided with a description the school setting and a description of the research

groups and participants. Presented next are individual case findings derived from surveys, laboratory assignments, and classroom assessments and observations. Chapter 5 subsequently summarizes individual case findings both within and between groups and provides observations and conclusions suggested by these analyses.

Instructional Context

The researcher instructed three classes of Advanced Placement Calculus AB during the Fall 2007. Two of the three classes were utilized to select participants in the study. Classes met five days a week for 52 minutes. The classroom was equipped with 30 laptop computers for student use.

Activities and lessons were developed concurrently with the course in response to classroom observations and general course flow. As such, the activities utilized had not been previously piloted. To provide an opportunity to refine activities prior to use in the research context, the first class of the day was utilized to pilot programming based activities. Although this class did not contain any participating students, it provided useful feedback from students and offered preliminary understanding of issues likely to be encountered by the participants. Additionally, I initially envisioned I would have a lunch period or a planning period in between the two classes under study in which to further reflect upon lessons. However, this was not possible due to scheduling considerations beyond my control.

The initial pool of potential participants consisted of 21 students in the programming class (case P) and 20 students in the non-programming class (case N). A third class was utilized as a pilot class consisting of 20 students who piloted programming based activities prior to their use with research participants in class P.

On the first day of class, an introductory survey was given to the students exploring *prior experience with technology* in the classroom, *computer programming experience, prior modes of classroom collaboration*, and *perceived uses of technology* in the mathematics classroom.

During the next three weeks, student and parent consent and assent was requested, student pairings where performed, and students computer accounts were established. This initial time period was used to establish a base conception of the limit concept. Although students did not directly utilize Maple during this period, they gained familiarity with the system through classroom demonstrations and explorations performed by me.

The semester's instructional activities began with a discussion of limits. This section outlines the instructional methodology used introduce the limit concept leading to an initial conception of limit. Computer laboratory activities began after this initial instructional phase further refining this initial conception. The students' initial conception was explored on the first quiz and provided a conceptual base from which the students' conceptual growth could be explored.

The authors of the course text give the formal definition of limit shown in Figure 10. Yet, typical in reform-based texts, this formal and very abstract notion is not

Definition of Limit

Let c and L be real numbers. The function f **has limit** L **as** x **approaches** c if, given any positive number ε, there is a positive number δ such that for all x,

$$0 < |x - c| < \delta \quad \Rightarrow \quad |f(x) - L| < \varepsilon$$

We write

$$\lim_{x \to c} f(x) = L$$

Figure 10. The formal definition of Limit

specifically explored. Instead, further exploration of this concept is relegated to the appendix of the text. Both secondary and post-secondary students have traditionally had tremendous difficulty understanding this formal notation-laden definition. Thus, most modern texts tend to focus on introducing the concepts behind the definition through specific, concrete examples usage in appropriate contexts. Subsequently, the concept can be refined in its meaning and interpreted with increased formality with or without (but typically without) reference to this formal definition. With this definition as the instructional goal, the laboratory activities focused upon promoting the development of this formal definition using the APOS conceptual framework of the limit coupled with specific software tools.

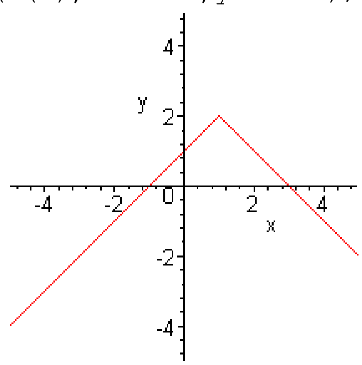
As a first, informal, step toward understanding limits, students learned to find limits graphically. Simultaneously, Maple is used as a demonstration tool to introduce the CAS environment and to promote key *process* and *object* perspectives with each of the two groups. In classroom discussion, the instructor utilized Maple to define piecewise functions and explore the graphical notion of limit at finite points. In each of the two classes, the notion of function was instantiated in two conceptually different ways. In case N, Maple’s mapping notation was utilized. This notation aligns closely with the standard textbook definition of a function that students have previously been introduced. It promotes a function-as-object conception of function. In case P, a procedure (in the computer science sense) was written implementing a functionally identical process. This approach, however, promoted a function-as-process view of functions. As the semester progressed, these two modes of function definition were adhered to within each respective class. Shown in Figure 11 is an example of this approach.

Non-Programming Groups, N_i

```
> f:=x->piecewise(x<1,x+1,x>=1,
-x+3):
> f(x);
```

$$\begin{cases} x+1 & x<1 \\ -x+3 & 1 \le x \end{cases}$$

```
> plot(f(x),x=-5..5,y=-5..5);
```



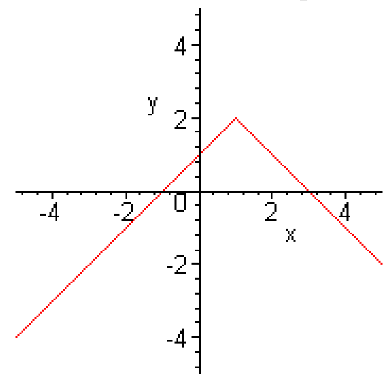

```
> limit(f(x),x=1);
```

2

Programming Groups, P_i

```
> with(CalcToolbox):
> f := proc( x )
   if x < 1 then
        x+1;
    else
       -x+3;
    end if:
end proc:
```

```
> plot('f(x)',x=-5..5,y=-5..5);
```




```
> lim('f(x)',x=1);
```

2.000000000

Note: the programming group used a different limit procedure, `lim`, than the non-programming group, `limit`. This was necessary as Maple's built-in limit function, `limit`, does not function properly with functions defined as procedures. Thus, a replacement limit function was developed so as to support the desired *function-as-process* perspective.

Figure 11. Formulations of function definition. The non-programming groups (left) utilized a traditional mapping formulation and the programming groups (right) utilized a procedural formulation of function definition.

After graphical exploration, a traditional exploration of indeterminacy ensues in concert with the development of standard *algebraic* methods for removing these indeterminacies (i.e. factoring, conjugate method, etc.). The formal notion of limit was not explicitly introduced until the fourth and final lab activity.

After seeing Maple as a demonstration tool by the instructor for a week, students were randomly paired and given opportunity to experiment with Maple. At this point, the

final research pairings were not available. Students were given an assignment introducing the idea of pair-programming and its associated roles (see Appendix A). As an introduction to the strategy, students were given a canned introduction to Maple to experiment with using the pair-programming model. Every 15 minutes, the instructor reminded the group member to swap roles. This provided a first introduction to the collaborative model they would use throughout the semester.

Initial Survey

Participants were surveyed at the beginning of the course to better understand their prior exposure to technology and collaboration in the mathematics classroom. Their primary exposure to technology in the classroom was consistently the graphing calculator. Participants in both groups saw the primary role of such technology as a labor saving device. Technology provided the ability to effortlessly produce graphical representations of functions and data and to perform complex arithmetic computations. One participant in class P saw the potential for technology to foster collaboration via online instruction. None described or perceived of such technology as an exploratory tool.

Additionally, no participant expressed prior experience with a computer algebra system and only two students, one in group N_2 and one in group N_3, expressed prior experience programming. All students indicated they had worked collaboratively in prior classes to complete and check homework. Only one participant in group N_3 indicated previous collaboration on a science project and on a geometry project. Thus, participants had very limited exposure to modern collaborative strategies within or outside the classroom.

Initial Limit Conception

Following the traditional instructional sequence just described, participants were asked, on a classroom assessment, to select the most appropriate definition of a limit from collection of common limit conceptions derived in a prior study of Williams (1991). Figure 12 presents the initial conception for each participant.

Initial conceptions appeared to be quite uniform across groups. The two most common limit conceptions for both groups are the dynamic-theoretical option followed by the acting-as-a-boundary option. These conceptions are consistent with the initial instructional sequence. At this point in the study, students had been shown traditional methods of understanding and calculating limits using graphical and algebraic strategies. The laboratory sequence had not commenced at this point.

The dynamic-theoretical choice reflected a perception derived from the graphical

Group		**Limit Conception**
P_1	H	1
	H	1
P_2	H	1
	L	3
P_3	H	2
P_4	H	2
N_1	L	4
	L	1
N_2	H	1
	L	2
N_3	H	1
	L	1
N_4	L	2
	L	1

Conceptions of Limit (Williams, 1991)

1) (*Dynamic-theoretical*) A limit describes how a function moves as x moves towards a certain point.
2) (*Acting as a boundary*) A limit is a number or point past which a function cannot go.
3) (*Formal*) A limit is a number that the y-values of a function can be made arbitrarily close to by restricting x-values.
4) (*Unreachable*) A limit is a number or point the function gets close to but never reaches.
5) (*Acting as an approximation*) A limit is an approximation that can be made as accurate as you wish.
6) (*Dynamic-practical*) A limit is determined by plugging in numbers closer and closer to a given number until the limit is reached.

H = Higher-performing, L = Lower-performing, X = no response

Figure 12. Initial Conception of Limit

introduction to limits. In exploring limits graphically, participants were told a limit is "the value a function intends to have at a given point." In order to determine this point, participants were instructed to imagine two cars driving along a road (the functions graph) each approaching a given location (x-value) from both directions (above and below the given x-value). The intended value of the function is the y-value of the destination of the two cars. This analogy clearly promotes a dynamic perception as it involves motion. The acting-as-a-boundary perception is also consistent with this dynamic-theoretical conception in the context of the algebraic strategies learned, i.e. factoring, conjugate method etc. In particular, each of these algebraic strategies yields a definite value corresponding to a definite location the two cars approach. The destinations are naturally conceived of as boundaries that the cars will not pass.

Lab 1 Results

The first Maple lab took place at the end of the third week. Students learned to define, evaluate, and plot functions in the Maple computer algebra system (CAS) using the respective mapping (class N) or procedural (class P) perspective and gained experience utilizing the pair-programming collaborative model. In this first lab experience, the researcher diligently directed students to change roles every 15 minutes stressing the importance of the specific individual roles in pair-programming. This lab specifically addressed parts one and two of the APOS decomposition for understanding limits (see Figure 7) by providing evidence of the student's ability to define and evaluate functions. A copy of the laboratory assignment is provided in the appendices.

Group P_1

As elaborated in Figure 13, the group properly defined functions, performed evaluations, and produced graphs of these functions. When producing a graph they failed to make use of the function definition itself suggesting that the purpose of defining a function is unique to the evaluation process. Thus, this group tends to see the process of function definition as specific to the process of evaluation only.

Additionally, in Figure 13, they were able to verbally describe how function evaluation is performed using a graph suggesting an understanding of graphical function evaluation. When asked how a particular right-sided limit is determined (see Figure 15), the students indicated an appropriate process for estimating the limit by approaching from both the left and right of *x=3*. Notably, the students did not make the

Problem #1:
Consider the following function f(x)= x-1
1a) Write a sequence of statements in Maple that implement this function.
 	
`> f := proc(x)`
`x-1;`
`end proc:`
	1b) Have Maple evaluate the function at x=3.5,3.25,3.1,3.01,3.001
	
	`> f(3.5); f(3.25); f(3.1);`
	`f(3.01); f(3.001);`
	`2.5`
	`2.25`
	`2.1`
	`2.01`
	`2.001`

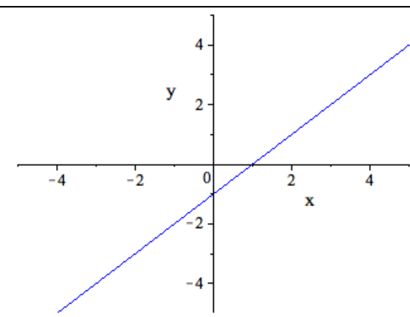
1d) Produce a plot in a [-5,5] by [-5,5] window
 	
`> plot (x-1, x=-5..5, y=-`
`5..5, color=blue);`
	

Figure 13. Group dialog with Maple CAS for Group P_1, Lab 1, Problem 1(a,b,d)

e) Explain how, using the graph, you evaluate f (4). *Answer: By looking at the x value you can see which y value coordinates with x=4, which is 17.*

Figure 14. Group P_1's response to Lab 1, Problem 1(e)

observation that (1c) was unnecessary thus failing to appreciate the one-sided nature of the question.

Notably on problem 3d (see Figure 17) they correctly deduced one-sided limits and justified their answers using a sequential argument. However, their response suggests some confusion in their conceptualization of the domain and rage processes taking place in the limiting process. The phrase "*As lim h(x) approaches 2 it does not exist…*," while not a specific instructional goal in this initial lab, does suggest a conceptual development to be made in future labs. When asked about the corresponding two-sided limit in 3e, they correspondingly utilized a sequential argument to explain the limit as x approaches 2 does not exist and, additionally, elaborated that these trends specifically implied an infinite discontinuity at x=2.

Problem 4 (see Figure 16) demonstrated a realization that the function definition's scope extended beyond mere evaluation. When asked to define the function k(x), the pair appropriately used the prior definitions of functions f and g. That is, there definition was specifically in terms of f and g as shown below. Also, in graphing the functions f and g

1f) Using the work you performed in steps (a-e) approximately what is lim f(x) as x approaches 3 from the right? Explain how you are estimating this; i.e. indicate specifically which of (a-e) you are basing your estimate on. *Answer: The limit is 2 by looking at b and c. The values approach 2 from both the negative and the positive side.*

Figure 15. Group P_1's response to Lab 1, Problem 1(f)

3d) Estimate the following limit, lim h(x) as x approaches 2 from the right (+) and lim h(x) as x approaches 2 from the left (-) and explain how you are estimating these limits.

Answer: As x approaches 2 from the left the limit is –infinity and as x approaches 2 from the right it is +infinity using b) and c)

3e) Do you think lim h(x) as x approaches 2 exists? Why?

Answer: As lim h(x) approaches 2 it does not exists because it is infinite discontinuity. You can see that all the numbers keep getting larger or smaller.

Figure 17. Group P_1's response to Lab 1, Problem 3(d,e)

(4see d), the prior definitions were employed. Unfortunately, the students did not indicate how these individual graphs could be used to perform evaluations of k.

This problem is where the groups work ends. The students did not respond to Problem 5 or 6 suggesting that time might have been a limiting factor. An action level of understanding outlined in steps one and two of the APOS decomposition of limit of function was clearly demonstrated by these students. With regard to implementation of the pair-programming model, the group of two high performing females worked quite

4) Consider the function k(x)= g(f(x)). (function f and g from #1, 2)
a) Write a sequence of statements in Maple that implement this function.

```
> k := proc(x);
g(f(x));
end proc:
```

b) Evaluate k(x) at x=0,1,2,3.

```
> k(0); k(1); k(2); k(3);
                                         2
                                         5
                                         8
                                        11
```

c) Explain in words what you think happens (what is the sequence of events) when you ask Maple to evaluate the expression g(f(2)).

Answer: It will plug 2 into the equation for f(x), then plus the answer from that into the equation for g(x).

d) Explain how, using ONLY the plots of f(x) and g(x), you could evaluate g(f(2)). Plot both graphs of f and g and explain how the two graphs could be used to evaluate g(f(2)).

```
> plot( [ f(x),g(x) ], x=-5..5, y=-5..5, color= [blue, black]);
```

Figure 16. Group P_1's response to Lab 1, Problem 4(a-d)

well together giving each other perfect evaluations. Both participants were able to seamlessly change roles and did not report any difficulties in utilizing the pair-programming paradigm. One group member did say she found the 15” display on the laptop to be small and difficult to see.

Group P_2

As with group P_1, this group was able to properly define and evaluate functions in problems one to four, but did not complete problems five and six. This group did demonstrate an understanding that function definitions extend beyond evaluation as they used prior definitions when producing graphs as well.

On problem 1b (see Figure 18), the group demonstrated an additional understanding not specifically addressed in the lab - an understanding of the distinction between a variable and a symbol, between a variable and a variables value. Note the pair first evaluated the function by assigning a specific value to variable x and evaluated the function using that variable. Subsequently, they removed the value attached to the symbol

(1b) Have Maple evaluate the function a x=3.5, 3.25, 3.1, 3.01, 3.001

```
> x :=3.5;
                                 x := 3.5
> f(x);
                                   2.5
> x:='x':
> f(x);
                                   x-1
> f(3.25);
                                  2.25
> f(3.1);
                                   2.1
> f(3.01);
                                  2.01
> f(3.001);
                                  2.001
```

Figure 18. Group P_2’s response to Lab 1, Problem 1(b)

x so that it was no longer a variable but a symbol; they confirmed this in the next statement when the functions evaluation at x resulted in a symbolic expression rather than a numeric value. While in the remainder of the problem they did not make subsequent use of this symbolic understanding, this observation suggests an ability to reason symbolically; that there is a difference between the symbol and the value it presently represents. Subsequently, they realize the value passed to the function need not be contained in a variable. Instead they could directly pass numeric value(s) to the function.

With problem 1f, shown in Figure 19, the group chose to use the functions value at x=3 to evaluate the one-sided limit suggesting that the limit is necessarily equal to the functions value at x=3. While this is true in the case of continuous functions, the group did not make reference to continuity in their answer. Further, even though the group performed evaluations necessary to correctly support their answer using a sequential argument, the group chose to appeal to functional evaluation at the point x=3.

The pair appropriately defined, analyzed, and responded to Problem 2 demonstrating an ability to perform graphical evaluation. On Problem 3 parts 3d and 3e (see Figure 20), rather than responding to the question using the specified evaluations to make a sequential argument, the pair appealed to prior algebraic understanding of rational functions. They reasoned that rational function h(x) possessed the asymptote x=2 as the denominator has a factor of (x-2) which does not cancel with a corresponding factor in the numerator. The pair circumvented a sequential argument by suggesting the limit was

(f) Using the work you performed in steps (a-e), approximately what is the limit of f(x) as x->3 from the right? Explain how you estimate this; i.e. indicate specifically which of (a-e) you are basing your estimate on.

*It is at 2 because at the point x=3, y=2. ^_^*

Figure 19. Group P_2's response to Lab 1, Problem 1(f)

(3d) Estimate the following limit, limit of h(x) as x->2 from the right and the limit of h(x) as x->2 from the left and explain how you are estimating these limits.

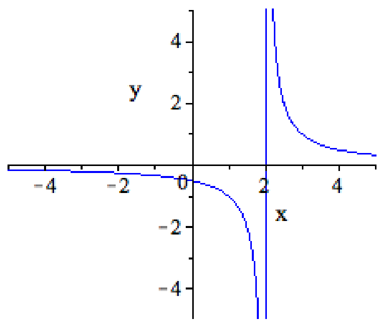
There is an asymptote at x=2, therefore the limit is +or- infinite.

(3e) Do you think the limit h(x) as x-> 2 exists? Why?

NO!!! Because there is an asymptote at x=2.

(3f) Produce a plot in a [-5,5] by [-5,5] window in blue.

```
> plot(h(x),x=-5..5,y=-5..5,color=blue);
```



(3g) Explain how the plot supports your answer to d and e.

Because x cannot equal 2.

Figure 20. Group P_2's response to Lab 1, Problem 3(d-g)

infinite since the functions graph possessed a vertical asymptote. They further suggest, in 3g, since h cannot be evaluated at x=2, this implies the limit as x approaches 2 does not exist.

On Problem 4 the group appropriately utilized prior definitions of functions f and g in the construction of function k and accurately explained how Maple would evaluate such an expression. However, in step 4d (see

Figure 21) the group demonstrated an inability to graphically evaluate a composition of two functions. Coupled with the observation that the pair could properly evaluate a single function graphically (see 1e in

Figure 21), this suggests the pair does not yet possess an object-level understanding of the graphical evaluation process.

<table>
<tr><td colspan="2">(1e) Explain how, using the graph, you evaluate f(4).

Go to point on the line that x=4, look at y axis, and there you go!

(4d) Explain how, using only the plots of f(x) and g(x), you could evaluate g(f(2)). Plot both graphs of f and g and explain how the two graphs could be used to evaluate g(f(2)).</td></tr>
<tr><td><code>> plot([f(x), g(x)] , x=-5..5, y=-5..5, color=[blue,green]);</code>

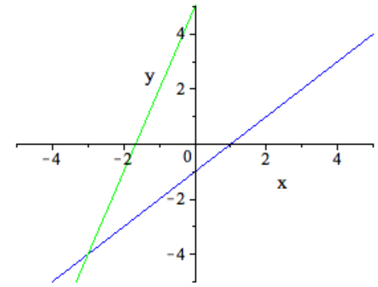
If you go to x=2, two points are possible outcomes. ?????????!?!!?!?!</td><td>
</td></tr>
</table>

Figure 21. Group P_2's response to Lab 1, Problems 1(e) and 4(d)

This group did demonstrate an action level of understanding of function evaluation, APOS step 1 and 2. Their reticence to make sequential arguments, coupled with their use of evaluation in problem one, suggests the pair believes that limits are evaluated using evaluation and that if such an evaluation can't be made, the limit must not exist.

As a pair, the group worked well together giving each other perfect peer reviews. Both expressed enjoyment of the lab and comfort with the partnership. Both individuals readily transitioned roles during the lab and demonstrated a clear understanding of the respective roles.

Group P_3 (singleton)

This group was able to properly define and evaluate functions f, g, h, and k in Problems 1 through 4 using algebraic and graphical means. However, after defining function f in Problem 1 (see Figure 22), the group attempted to redefine the function using notation to which they are more accustomed. They subsequently realized the initial definition applied, and did not attempt further redefinition. As with group P_1, the group did not make use of function definitions when producing graphs, instead they simply entered the expressions for the functions as exemplified in Figure 23 .

On problem 1f (see Figure 24), the pair explained the limiting behavior using the graph and evaluation and did not make any supported sequential argument. Instead they relied upon the graph to ascertain limiting behavior. Additionally, the pair only considered the behavior of the function on the right side, however. The two-sidedness of the limit definition was not completely understood. The answer further suggested since the value at x=3 is 2, the limiting value must be 2; specifically, the limit as x approaches 3 is dependent upon the value of the function at x=3. This response further suggested the function must have a value at x=3 in order for the limit to exist.

Problem #1: Consider the function f(x)=x-1

(a) Write a sequence of Maple statements that implements this function.

```
> f := proc(x)
x-1;
end proc:
```

(b) Have Maple evaluate the function at x= 3.5, 3.25, 3.1, 3.01, 3.001

```
> x := 3.5;
                                  x := 3.5
> f(x)= x-1;
                                  2.5 = 2.5
> f(x)=x-1;
                                  2.5 = 2.5
> f(3.5);
                                     2.5
> x:='x';
                                   x := x
> f(3.25);
                                    2.25
> f(3.1);
                                     2.1
> f(3.01);
                                    2.01
> f(3.001);
                                    2.001
```

Figure 22. Group P_3's response to Lab 1, Problem 1(a,b)

(1d) Produce a plot in a [-5,5] by [-5,5] window.

```
> plot(x-1,x=-5..5,y=-5..5,color=blue);
```

Figure 23. Group P_3's response to Lab 1, Problem 1(d)

(f) Using the work you performed in steps (a-e), approximately what is the limit of f(x) as x approaches 3 from the right? Explain how you are estimating this; i.e. indicate specifically which of (a-e) you are basing your estimate on.

The limit is 2 as it approaches 3 from the right according to the graph. Also, according to the algebra,

```
> f(3);
                         2
```

since the limit approaches 3 from the right we know that +2 is the correct limit, in addition to using our graph to verify.

Figure 24. Group P_3's response to Lab 1, Problem 1(f)

On problem 3d (see Figure 25), the group demonstrated confusion of the domain and range processes involved in determining the limit. The pair correctly inferred the graphs behavior on both sides of the asymptote x=2. They suggested the limit of the function is equal to the domain value being approached x=2 rather than the functions value as x approaches 2. The response further suggested, consistent with their initial limit conception, the limit is a value that is approached but never attained, i.e. it is unreachable. In contradiction to their analysis in 3d, they argued in 3e that the limit does not exist. Clearly, the group had difficulty inferring limiting behavior using a sequential argument but was able to correctly infer such behavior from the functions graph. This observation is further supported by the following response to part 3g (see Figure 26) where it was claimed the domain value x=2 is never reached thus there is no limit. There was no discussion of the range process.

This pair did demonstrate an action-level of understanding of the process of function evaluation, APOS Step 1 as well as action level of understanding of Step 2. However, their understanding of Step 2 has not been interiorized as evidenced by their lack of use of these evaluations to justify limiting behavior. As this student was a singleton, peer reviews were not considered.

(d) Estimate the following limit, the limit of x as it approaches 2 from the right and the limit of x as it approaches 2 from the left and explain how you are estimating these limits.

As x approaches 2 from the right the function increases, and as it approaches 2 from the left the function decreases. However, in neither instance does the limit reach 2, so the limit is 2.

(e) Do you think the limit of h(x) as it approaches 2 exists? Why?

No, it does not exist because there is a vertical asymptote and the values go on to negative and positive infinite.

Figure 25. Group P_3's response to Lab 1, Problem 3(d,e)

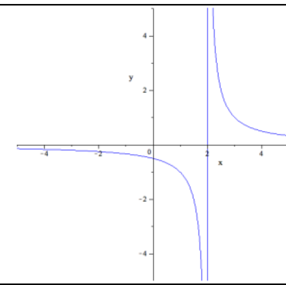
(g) Explain how the plot supports your answers to d and e. *You can see on the graph that neither the function approaching from the left nor the right reaches 2 because of the vertical asymptote.*	

Figure 26. Group P_3's response to Lab 1, Problem 3(g)

Group P_4 (singleton)

The group properly defined, evaluated, and graphed functions f, g, h, and k. As with the other groups, Problems 5 and 6 were not attempted. The group utilized the function definitions when graphing as well as when evaluating suggesting a broader perspective of the scope of function definitions.

In problem 1f (see Figure 27), the group determined the limit using a graphical evaluation. They made no reference to either of the two one-sided behaviors explored in parts b and c (not shown). This reliance on the ability to evaluate the given function was further reflected in the following response to problem 3 parts d and e (see Figure 28). The students correctly explained the one-sided behavior of function h near x=2 but suggested the reason the limit fails to exists is that the function is undefined at x=2. This is the only programming group that specifically describes the process of function composition in graphical terms (See Figure 29).

This group demonstrated an action-level of understanding of function evaluation and an ability to perform a sequence of coordinated evaluations around a given point. Thus, APOS steps 1 and 2 were attained. Like other groups, however, they avoided making sequential arguments when ascertaining or supporting proposed limiting behavior.

1f.) Using the work you performed in steps (a.-e.), approximately what is the limit as "x" approaches 3 from the right of f(x)? Explain how you are estimating this; i.e. indicate specifically which of (a.-e.) you are basing your estimate on.

Finding where x=3 and following it to the "y" value, like 1e.

Figure 27. Group P_4's response to Lab 1, Problem 1(f)

3d.) Estimate the following limit: the limit as x approaches 2 from the right of h(x), and the limit as x approaches 2 from the left of h(x).

from the right, the limit is positive infinity, from the left, the limit is negative infinity. As x approaches from the right, x will always be a little greater than 2, and approaching from the left, x will always be a little less than 2.

3e.) Do you think the limit as "x" approaches 2 of h(x) exists? Why?

No, because when 2 is plugged into the equation the answer is 1/0, which means it does not exist.

Figure 28. Group P_4's response to Lab 1, Problem 3(d,e)

4d.) Explain how, using ONLY the plots of f(x) and g(x), you can evaluate g(f(2)). Plot both graphs of f and g and explain how the two graphs could be used to evaluate g(f(2)).

Find 2 on the f(x) line and the value of it is 1. Then find what g(x) equals at x=1.

```
> plot ( [ g(x), f(x)], x=-5..5,
y=-5..5, color=[green,blue]);
```

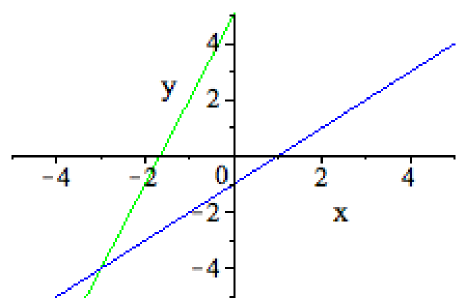


Figure 29. Group P_4's response to Lab 1, Problem 4d

Group N_1

This group was able to correctly define functions f, g, h, and k and properly evaluate and plot functions f, g, and h using these definitions; Function f's definition and evaluation is highlighted in Figure 30. Function k was defined properly but not evaluated.

The pair was able to properly describe a process for evaluating a function graphically in 1e as well as estimate a one-sided limit in 1f (see Figure 31). The explanation, however, relied upon the graph rather than a sequential argument involving 1b and 1c. Significantly, in Problem 3, a sequential argument was given coupled with a graphical justification (See Figure 32). They argued the limit was non-existent in 3e as a result of a vertical asymptote x=2. Their response suggested that the pair may have some comfort with a sequential limiting argument in 3d but ultimately they supported the conclusion with a graphical justification in 3e.

The pair demonstrated an action-level of understanding of function evaluation using a functions definition as well as using a functions graph. The group was also able to perform a sequence of evaluations progressively closer to a given point. Thus the group attained an action level of understanding as outlined in APOS steps 1 and 2. The reticence to make sequential arguments, however, suggested that step 2 has not been interiorized as an object as yet.

During the lab, the group demonstrated a maturing ability to reason about limits using sequential arguments. The group did not complete problems four through six, however. Both members gave each other perfect peer reviews but one expressed difficulty changing roles indicating that the exchange added confusion to the activity.

Problem #1: Consider the function f(x)=x-1
(a) Write a sequence of Maple statements that implement this function.

```
> f := x->x-1;
                                        f := x -> x - 1
```

(b)Have Maple evaluate the function at x=3.5,3.25, 3.1, 3.01, 3.001

```
> f(3.5);
                                             2.5
> f(3.25);
                                            2.25
> f(3.1);
                                             2.1
> f(3.01);
                                            2.01
> f(3.001);
                                            2.001
```

Figure 30. Group N_1's response to Lab 1, Problem 1(a,b)

(1e) explain how, using the graph, you evaluate f(4).

you move along the x-axis until you reach 4, and look up at the y-value until you meet the line.

(1f) Using the work you performed in steps (a-e), approximately what is limit of f(x) as x->3 from the right? Explain how you are estimating this; i.e. indicate specifically which of (a-e) you are basing your estimate on.

The limit is 2. Move along the x-axis from the right side to the left until u get to 3. Then move up until u reach the line of the function, estimate the y value. This estimate is based on (e).

Figure 31. Group N_1's response to Lab 1, Problem 1(e,f)

(3d) Estimate the following limit, limit of f(x) as x->2 from the right and the limit of f(x) as x->2 from the left, and explain how you are estimating these limits.

From the right the limit is positive infinity because the function gets bigger and bigger. From the left, the limit is -infinity because the function get smaller and smaller.

(3e) Do you think the limit of f(x) as x->2 exists? Why?

No. Because there is a vertical asymptote at x=2, and it is infinitely discontinuous.

(3g) Explain how the plot supports your answer.

There is an asymptote at x=2 because 2 would make the domain 0, and also the values from the right reach infinity, while the values from the left reach negative infinity.

Figure 32. Group N_1's response to Lab 1, Problem 3(d,e,g)

.

Group N_2

On Lab 1, the group was able to properly define functions and perform evaluations. Like many other pairs, when plotting graphs they did not make use of the previous function definition suggesting the sole purpose of defining a function was to permit evaluation; the scope of a function's definition was not perceived to extend to other more situations such as graphing. This perception persisted until Problem 4 when they realized prior definitions can be generally utilized; when forming a new function which required the composition of two prior functions they correctly defined the function using composition of prior functions.

The pair verbally described how function evaluation was performed using a graph. When asked how a particular right-sided limit is determined the students indicated an appropriate process for estimating the limit and correctly referenced the appropriate sequence of function evaluations given in 1b (See Figure 33). By not referencing (1c), the group clearly demonstrated an understanding of the one sided nature of the question.

On problem 3d (See Figure 34), they correctly deduced one-sided limits and justified their answer using a sequential argument. In Problem 3e, they correctly suggested the limiting behavior near x=2 implied a vertical asymptote (rather than the asymptote being the cause of the infinite behavior). This group clearly demonstrates an action level of understanding of function evaluation as well as outlined in the APOS decomposition, steps 1 and 2.

With regard to implementation of the pair-programming model, the group found switching roles beneficial in keeping them on task but indicated Maple was difficult to "get used to." This was significant as on the preliminary lab experience, these two had

some difficulty staying on task. As it turned out, this pair had been good friends for many years and tended to have more casual interactions; one member also had a tendency to provide humorous answers to questions often at the expense of answering the question. Not surprisingly, the pair gave each other perfect scores on their peer review following the first lab.

(1b) Have Maple evaluate the function at x=3.5, 3.25, 3.1, 3.01, 3.001.

```
> f(3.5); f(3.25); f(3.1);f(3.01);f(3.001);
                                    2.5
                                   2.25
                                    2.1
                                   2.01
                                   2.001
```

(1f) Using the work you performed in steps (a)-(e), approximately what is the limit x->3+ f(x)? Explain how you are estimating this; specifically which of (a-e) you are basing your estimate on.

(b) shows us that as x->3+, f(x) gets closer and closer to 2. The graph (d) confirms this pattern too.

Figure 33. Group N_2 's response to Lab 1, Problem 1(b,f)

(3b) Have Maple evaluate the function at x=2.5, 2.25, 2.1, 2.01, 2.001.

```
>  h(2.5);h(2.25);h(2.1);h(2.01);h(2.001);
                                2.000000000
                                4.000000000
                                    10.
                                   100.
                                   1000.
```

(3c) Have Maple evaluate the function at x=1.5, 1.75, 1.9, 1.99, 1.999.

```
> h(1.5);h(1.75);h(1.9);h(1.99);h(1.999);
                               -2.000000000
                               -4.000000000
                                   -10.
                                   -100.
                                  -1000.
```

(3d) Estimate the following limit, lim x->2+ h(x) and lim x->2- h(x) and explain how you are estimating these limits.

From b and c, Infinity from the right; as x->2 h(x) approaches infinity. -Infinity from the left; as x->-2 h(x) approaches -infinity.

Figure 34. Group N_2 's response to Lab 1, Problem 3(b-d)

Group N_3

The pair was able to properly define functions, perform evaluations making effective use of previous function definitions to evaluate and graph. On 1e (See Figure 35), the pair was able to verbally describe how function evaluation is performed using a graph as well. When asked how a particular right-sided (1f) limit is determined, rather than use a graphical or sequential argument, the pair used the algebraic method of substitution.

In (3d), they described appropriate limiting behavior but did not indicate how this was inferred. Thus, it appears that student a more confident with algebraic methods involving predetermined steps than with sequential behavior. Moreover, in 3e, they indicated the limit did not exist because of a vertical asymptote $x = 2$.

An action-level understanding of function evaluation was clearly demonstrated in this lab. However, as with other groups, this group tended to stick with algebraic and graphical justifications over sequential arguments. The pair additionally demonstrated an action level of understanding of the process of evaluation at successively closer points, APOS step 2. However, it does not appear that at this point step 2 has been interiorized to an object. This group also appears to have had time issues, and thus did not complete much beyond Problem 3 of the six problems.

With regard to implementation of the pair-programming model, both members indicated that the pair-programming strategy was enjoyable; “This Rocks!” stated the lower performing group member. The members each gave themselves perfect peer reviews. The pair had no difficulty changing roles and, in fact, was consistently very aware of when the next transition was to occur; they did not require prompting from me.

(1e) Explain how, using the graph, you evaluate f(4).

You go to 4 on the x axis and go straight up til you get to the line. Then you find the y value that is at that same level.

(1f) Using the work you preformed in steps (a-e), approximately what is lim f(x)?
Explain how you are estimating this; i.e. indicate specifically which of (a-e) you are basing your estimate on.

The answer is 2 by using the method of substitution.

Figure 35. Group N_3's response to Lab 1, Problem 1(e-f)

Group N_4

This group of two lower performing males properly defined functions f, g, h, and k. The defined function was, however, initially used solely for evaluation. When graphing the function f, the group did not utilize the prior function definition and instead typed in the expression to be graphed directly as evidenced in f parts 1a and 1d (see Figure 36).

1) Consider the function f(x)= x-1
1a) Write a seq. of statements in Maple that implements this function.

```
> f := x-> x-1;
```

1b) Have Maple evaluate this function @ x=3.5, 3.25, 3.1, 3.01, and 3.001.

```
> f(x);
```

$$x - 1$$

```
> f(3.5);
```

$$2.5$$

```
> f(3.25);
```

$$2.25$$

```
> f(3.1);
```

```
> f(3.01);
```

$$2.01$$

```
> f(3.001);
```

$$2.001$$

1d) Produce a plot in a [-20,20] by [-20, 20] window in blue.

```
> plot( x-1, x=-20..20, y=-20..20, color=blue);
```

Figure 36. Group N_4's response to Lab 1, Problem 1(a,b,d)

Eventually, the pair came to understand function definitions applied more generally and could be utilized as shown in problem 2d (see Figure 37) to produce graphs. This generality evidently ended when it came to function composition (see Figure 38), however, as in the case of function k in 4a, the pair performed the composition themselves, i.e. $g(f(x)) = 3f(x) + 5 = 3(x-1) + 5 = 3x + 2$, rather than have Maple do so (see Figure 38). The pair has a somewhat domain-specific understanding of function definition although they demonstrated increasing understanding.

Additionally, the pair appears to be developing a clearer understanding of graphical evaluation. When describing how to perform graphical evaluation (see Figure 39), the pair initially relied upon evaluation rather than the graph as evidenced below by the wording of their response to 1e and 1f. Problem two (see Figure 40) provided the first verbal hint of understanding graphical function evaluation.

Problem 3 (see Figure 41) contained the first attempt at a sequential argument. In 3d, the pair demonstrated confusion with the domain and range processes involved claiming that the limit was 2 as we approached 2 when in fact the limits were positive and negative infinity as suggested by the sequence of evaluations made in 3b and 3c. Then in 3e, they claim that the limit does not exist as substitution does not yield and answer.

2d) Produce a plot in [-5,5] and [-10,20] window in Black.

```
> plot(g(x),x=-5..5, y=-10..20,color=black);
```

Figure 37. Group N_4's response to Lab 1, Problem 2d

4) Consider the function k(x)= g(f(x)), plug in f for g.
4a) Write a seq. of statements in Maple that implements this function.

```
> k:= x -> 3*x+2;
```

Figure 38. Group N_4 's response to Lab 1, Problem 4(a)

1e) Explain how, using the graph, you evaluate f(4).

When x=4, y values = 3, because when you plug in 4 for x, the y-value equals 3.

1f) Using the work you preformed in steps (a-e), approximately what is lim f(x)?
Explain how you are estimating this; i.e. indicate specifically which of (a-e) you are basing your estimate on.

Using substitution, when you plug in 3 to the formula x-1 you get 2, and we checked it with the graph in part 1d.

Figure 39. Group N_4 's response to Lab 1, Problem 1(e-f)

2e) Explain how, using the graph, you evaluate g(4).

When you look at the input of 4 for x-values, you can see that the y-values go up to 17.

Figure 40. Group N_4 's response to Lab 1, Problem 2(e)

3b) Have Maple evaluate the function @ x= 2.5, 2.25, 2.1, 2.01, 2.001.

```
> h(2.1); h(2.01); h(2.001);
                                   10.
                                   100.
                                   1000.
```

3c) Have Maple evaluate the function @ x= 1.5, 1.75, 1.9, 1.99, 1.999.

```
> h(1.9); h(1.99); h(1.999);
                                   -10.
                                   -100.
                                   -1000.
```

3d) Estimate the following Limit, lim x-> 2+ and lim x-> 2- and explain how you are estimating these limits.

For the limit of x-> 2 from the right as x values approach 2 we can substitute values in to our equation for values close to 2 and see a limit of 2, and as x values approach 2 from the left, a limit can be found at 2.

3e) Do you think that lim x-> 2 of h(x) exists?

No, b/c when you plug in 2 you get an indeterminate form.

Figure 41. Group N_4 's response to Lab 1, Problem 3(b-e)

On Problem 4 (see Figure 42), the students correctly described a process graphically evaluating the composition of a function given individual graphs of f and g. The pair clearly possessed an action level of understanding of function evaluation both utilizing a functions algebraic definition as well using its graph, i.e. APOS step 1. However, while an action level of understanding of APOS step 2 was in evidence, it does not appear that step 2 has yet been interiorized to an object.

The pair worked well together and gave one another perfect peer reviews. One member suggested that it would be advantageous to determine who works best at the computer and minimize changes. The pair clearly was hesitant to switch roles and saw having specific predefined jobs would be more beneficial- ostensibly faster. This student was much more comfortable (driving) typing at the computer rather at the broader high level planning role entailed in the navigator role.

d) Explain how, using ONLY the plots of f(x) and g(x), you could evaluate g(f(2)). Plot both graphs of f and g and explain how the two graphs could be used to evaluate g(f(2)).

You look at f(x) graph, identify where x values = 2 and find the y value then use that y value for g(x) as your x and find g(x)'s y value at that x value.

Figure 42. Group N_4 's response to Lab 1, Problems 4(d)

Lab 1 Summary

Participants learned to define, evaluate, and plot functions in the Maple computer algebra system using their respective notations - mapping (class N) or procedural (class P). The specific intent of this lab was to address parts one and two of the APOS decomposition. All groups attained the intended action level of understanding of function evaluation as well as an ability to perform a sequence of coordinated evaluations. The progression towards a complete understanding of limit within the APOS decomposition is shown in Figure 43. The vertical line highlights the intended level of attainment within the decomposition at the completion of the lab and the shading indicates the group's actual level of attainment.

In addition to targeting the ability to systematically evaluate a function around a given point, this lab further intended to provide participants the opportunity to make inferences regarding limiting behavior utilizing sequential arguments by fostering an interest in a systematic domain process. Five of the eight groups, $P_2, P_4, N_1, N_3,$ and N_4, failed to make use of readily available data to construct sequential arguments for limiting behavior. Instead, these groups relied upon graphical and algebraic methods. The remaining three groups did offer at least one sequential argument. Of the two classes, P and N, the non-programming groups appeared less willing to offer such sequential support.

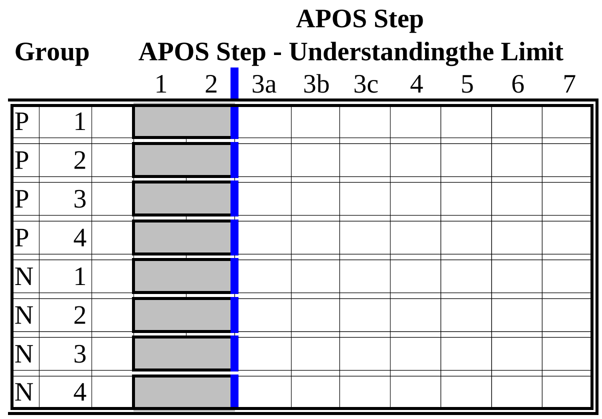
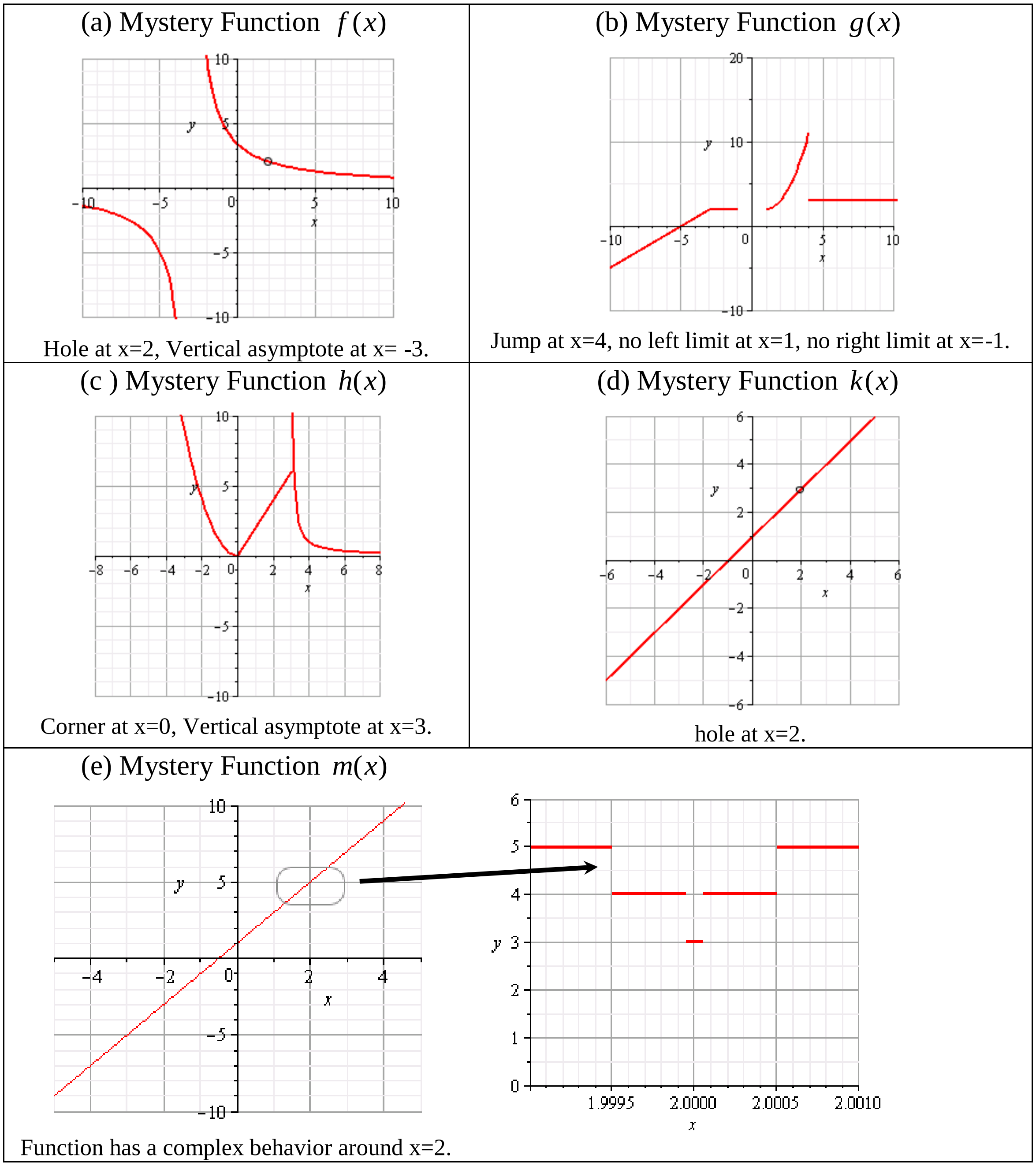


Figure 43. APOS Level of Attainment Following Lab 1. The shaded region represents the actual attainment and the vertical line indicates the intended attainment level.

Lab 2 Results

Lab 2 took place during the fourth and fifth week. Students in class N utilized and students in class P developed and utilized a software tool called `simpleLimitTable` to explore the behavior of five mysterious functions, shown in Figure 44.



Hole at x=2, Vertical asymptote at x= -3.

Jump at x=4, no left limit at x=1, no right limit at x=-1.

Corner at x=0, Vertical asymptote at x=3.

hole at x=2.

Function has a complex behavior around x=2.

Figure 44. Mystery Functions for Exploration by all groups in Lab 2

With this tool, students explored the behavior of each predefined mystery function near specified points. Use of this tool was intended to prompt a sequential understanding of limiting processes as outlined in steps 2, 3a, and 3b of the APOS framework for understanding limit.

Students in the class P were the additional task of developing the `simpleLimitTable` procedure rather than simply using a preconstructed version. The `simpleLimitTable` procedure took two parameters, a function, `f`, and a point, `x`, and displays an appropriate sequence of evaluations at points within 0.1, 0.01, and 0.001 units above and below point `x`. Shown in Figure 45 is a table produced for function $p(x) = x^2$ around the point x=1.

To guide the development of this procedure, a pre-lab activity was performed with the programming groups, the day prior, introducing the concepts of *procedure*, *local variable*, *program sequence*, and *function parameters* to *procedures*. In this activity, groups created a procedure for solving a quadratic equation, a well-practiced and familiar skill to students at this level. The non-programming groups were shown how to use a preconstructed version of the `simpleLimitTable` procedure. Both groups then explored the collection of mystery functions.

```
> p := proc(x) x^2; end proc:
> simpleLimitTable(p,1);
        (x-values)        p(x)
         .900000        0.810000
         .990000        0.980100
         .999000        0.9989001
        1.001000        1.002001
        1.010000        1.020100
        1.100000        1.210000
```

Figure 45. Sample Application of `simpleLimitTable` tool

The mysterious nature of these functions, shown in Figure 44, was that neither an analytic nor graphical representation of the functions were available to the students; the only tool of analysis was simple evaluation and the `simpleLimitTable` tool. Specifically, students did not have the graphs nor could they produce the graphs shown in Figure 44. The intent was to force students to pursue exploration of each graphs behavior at specific points using *sequential* explorations near a given point. Following Lab 1, students tended to avoid this approach opting to utilize graphical or analytic approaches to explanation. The lab addressed notions 1, 2, 3a, and 3b of the APOS decomposition by requiring function evaluation and by prompting students to interiorize the domain process of successive approximation nearer and nearer a given point.

Group P_1

In exploring function f near x=1, 2, and -3 (see Figure 46), the group performed a few evaluations on both sides of the points of interest and described the behavior of the function as decreasing as x approached the respective value x=1, 2, and -3, yet the sequence of evaluations clearly continue to increase beyond the respective x. No mention of specific limiting behavior is given. Additionally, the pair mistakenly uses the value x=3 rather than x=-3. It is significant the sequence of one-sided evaluations given in 2 and 3 are not presented in increasing (decreasing, respectively) order of x suggesting a lack of understanding of the domain process outlined in steps 2 and 3a (See Figure 7). In fact, they misinterpret the resulting output and indicate an increasing or decreasing trend as x approaches 1 when in fact the sequence of evaluation point does not approach 1. Their explanation is consistent with their initial limit conception that "*a limit describes how a function moves as x moves toward a certain point.*"

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."

```
> f(0); f(.9); f(1.01); f(1.1);
```

3.333333333

2.56102564

2.493765586

2.439024390

Description of behavior: as x approaches 1 the function decreases.

2) Explore and Describe the behavior of function f(x) at and near x=2.

```
> f(1.9); f(2.01); f(2.1);
```

2.040816327

1.996007984

1.960784314

Description of behavior: as x approaches 2 the function decreases.

3) Explore and Describe the behavior of function f(x) at and near x= -3.

```
> f(2.9); f(3.01); f(3.1);
```

1.694915254

1.663893511

1.639344262

Description of behavior: as x approaches -3 the function decreases.

Figure 46. Group P_1's response to Lab 2, Problems 1-3 exploring function `f(x)`

The pair produced the following response and tested their `simpleLimitTable` procedure on mystery function f in Figure 47 . The pair demonstrated appropriate use of parameters, an understanding of the relevant domain process, as well as proper use of the parameters and functions. In spite of their analysis for function f in Figure 46, the pair does show an awareness of the requisite domain process.

Definition of Procedure

```
> simpleLimitTable := proc( f, x )
      print(x+0.1,f(x+.1) );
      print(x+0.01,f(x+.01) );
      print(x+.001,f(x+.001) );
      print(x-0.001,f(x-0.001) );
      print(x-.01,f(x-.01) );
      print(x-.1,f(x-.1) );
end proc:
```

Trial Application of Procedure to Function f

```
> simpleLimitTable(f,3);
  3.1, 1.639344262
  3.01, 1.663893511
  3.001, 1.666388935
  2.999, 1.666944491
  2.99, 1.669449082
  2.9, 1.694915254
```

Figure 47. Group P_1's `SimpleLimitTable` Implementation and Demonstration

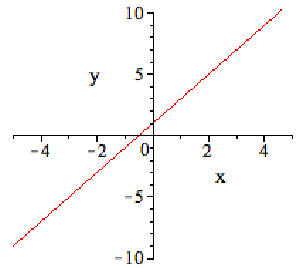
Using this procedure, the students focused their attention on understanding the behavior of functions g, h, k, and m. For functions g, h, and k, the pair appropriately applied their procedure but failed to state any conclusions about the limiting behavior of the functions suggesting they were more focused upon the development of the procedure rather than its subsequent use (see Figure 48).

Function m was a challenging exploration for the pair (see Figure 49). When asked to plot the function, the pair correctly produced a plot that appears linear. Then they applied `simpleLimitTable` to the graph at x=2 but again did not respond to the request to state the limiting value.

<table>
<tr><td colspan="2">Problem #1 - Exploration of g(x)
Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.</td></tr>
<tr><td><code>> simpleLimitTable(g,-1);</code></td><td><code>-0.9, undefined
-0.99, undefined
-0.999, undefined
-1.001, 2
-1.01, 2
-1.1, 2</code></td></tr>
<tr><td><code>> simpleLimitTable(g,1);</code></td><td><code>1.1, 2.01
1.01, 2.0001
1.001, 2.000001
0.999, undefined
0.99, undefined
0.9, undefined</code></td></tr>
<tr><td><code>> simpleLimitTable(g,4);</code></td><td><code>4.1, 3
4.01, 3
4.001, 3
3.999, 10.994001
3.99, 10.9401
3.9, 10.41</code></td></tr>
</table>

Figure 48. Groups P_1's exploration of function g(x) using the `simpleLimitTable` procedure

```
> plot ( m(x), x=-5..5, y=-10..10);
```




```
> simpleLimitTable(m,2);
2.1, 5.2
2.01, 5.02
2.001, 5.002
1.999, 4.998
1.99, 4.98
1.9, 4.8
```

Figure 49. Group P_1's exploration of m(x) near x=2 to a tolerance of 0.001

The pair was then given the task of refining the `simpleLimitTable` procedure by creating a new procedure, `simpleLimitTable2` that would approach a given point more closely. Specifically, the procedure evaluates the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point. The pair again produced an appropriate procedure and utilized it to again examine the behavior of function m near x=2. This resulted in a table that was discrepant with the table produced by `simpleLimitTable` (see Figure 50) suggesting a different limiting value.

Definition of Procedure

```
> simpleLimitTable2 := proc( f, x )
      print(x+0.1,f(x+.1) );
      print(x+0.01,f(x+.01) );
      print(x+.001,f(x+.001) );
      print(x+.0001,f(x+.0001) );
      print(x-.0001,f(x-.0001) );
      print(x-.001,f(x-.001) );
      print(x-.01,f(x-.01) );
      print(x-.1,f(x-.1) );
end proc:
```

Application of Procedure to Function m

```
> simpleLimitTable2 (m,2);
      2.1, 5.2
      2.01, 5.02
      2.001, 5.002
      2.0001, 4.
      1.9999, 4.
      1.999, 4.998
      1.99, 4.98
      1.9, 4.8
```

Figure 50. Group P_1's Implementation of `simpleLimitTable2` and application at x=2

This discrepancy was noted but the pairs explanation described the trend as the function no longer monotonically increased as x=2 was approached suggesting a focus upon the monotonicity of the range process rather than the limiting behavior.

`SimpleLimitTable2` was further refined, resulting in another new procedure, `simpleLimitTable3`, which would approach the given point more closely. Specifically, the procedure it evaluates the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point. The pair again produced an appropriate procedure and utilized it to again examine the behavior of function m near x=2 (see Figure 51). This resulted in a table that was again discrepant with the table produced by both `simpleLimitTable` and `simpleLimitTable2`.

Finally the group was asked to look as closely as necessary at the graph of m so as to explain the contradictory results produced by `simpleLimitTable`, `simpleLimitTable2`, and `simpleLimitTable3`. The pair produced the plot and explanation in Figure 52. The graph more clearly showed the unexpected behavior around x=2. From their response to Problem 8, the pair still appeared to hold the belief that the limiting value must depend upon the functions value at x=2.

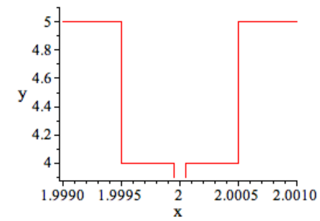
Definition of Procedure	**Application of Procedure to Function m**

`> simpleLimitTable3 := proc( f, x )`
`print(x+0.1,f(x+.1) );`
`print(x+0.01,f(x+.01) );`
`print(x+.001,f(x+.001) );`
`print(x+.0001,f(x+.0001) );`
`print(x+.00001,f(x+.00001) );`
`print(x-.00001,f(x-.00001) );`
`print(x-.0001,f(x-.0001) );`
`print(x-0.001,f(x-0.001) );`
`print(x-.01,f(x-.01) );`
`print(x-.1,f(x-.1) );`
`end proc:`
	`> simpleLimitTable3(m,2);`
	`2.1, 5.2`
	`2.01, 5.02`
	`2.001, 5.002`
	`2.0001, 4.`
	`2.00001, 3.`
	`1.99999, 3.`
	`1.9999, 4.`
	`1.999, 4.998`
	`1.99, 4.98`

Figure 51. Group P_1's implementation of `simpleLimitTable3` near x=2

<table>
<tr><td>

```
> plot ( m(x), x=1.999..2.001, y=3.9..5.1);
```



</td></tr>
<tr><td>8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

Plugging in the answer of the limit into the equation to make sure that the answers of the limit are the same.</td></tr>
</table>

Figure 52. Group P_1's response to Lab 2, Problem 8

To summarize, the pair was able to construct the necessary procedures and demonstrated an understanding of appropriate domain processes attaining understandings outlined in APOS steps 1, 2, and 3a but not 3b. While the group was able to infer some discrepant behavior using the tool, the group did not utilize the tool's output to justify or support any inferred limiting behavior. It appears, from their lack of response to the questions, a greater focus was placed upon the creation of the tool than on its application. There is a definite lack of understanding of coordinated domain and range processes. Little attention was paid to the range behavior of the given function. The pair continued to hold that a limits value is dependent upon the functions value at the limit point. As a pair, the group continued to perform well utilizing the pair programming model again giving each other perfect peer evaluations.

Group P_2

In exploring function f near x=1, 2, and -3, the group evaluated the function f at one unrelated point and gave no further response (see Figure 53). It is not clear why the group did not complete this part of the lab.

The pair successfully implemented the `simpleLimitTable` procedure and demonstrated its application to mystery function f as shown in Figure 54. The pair demonstrated appropriate use of parameters, an understanding of the relevant domain process, as well as proper use of the procedure. The group additionally experimented with variable names longer than a single character, e.g. variable `fred` in Figure 54, demonstrating an openness to experiment symbolically.

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."

```
>  f(10);
  0.7692307692
```

2) Explore and Describe the behavior of function f(x) at and near x=2.

```
>
```

3) Explore and Describe the behavior of function f(x) at and near x= -3.

```
>
```

Figure 53. Group P_2's response to Lab 2, Problem 1-3 exploring mystery function f(x)

Definition of Procedure	Application of Procedure to f

`> simpleLimitTable:=proc(fred, x)`
`  print( x+0.1, f(x+0.1) );`
`  print (x+0.01, fred(x+0.01));`
`  print (x+.001, fred(x+.001));`
`  print (x-.001, fred(x-.001));`
`  print (x-0.01, fred(x-.01));`
`  print (x-.1, fred(x-.1));`
`end proc:`
	`> simpleLimitTable(f,2);`
	`  2.1, 1.960784314`
	`  2.01, 1.996007984`
	`  2.001, 1.999600080`
	`  1.999, 2.000400080`
	`  1.99, 2.004008016`
	`  1.9, 2.040816327`
	`  2.9, 1.694915254`

Figure 54. Group P_2's `simpleLimitTable` implementation and demonstration

Using this procedure, the students focused attention on understanding the behavior of functions g, h, k, and m. Of functions g, h, and k, only function g's evaluation is shown in Figure 55. The pair appropriately applied the procedure but failed to state any conclusions about the functions. As with programming group P_1, this suggests greater focus upon development of the procedure than its subsequent use.

Problem #1 - Exploration of g(x) Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.	

`> simpleLimitTable(g,-1);`	`-0.9, undefined`
	`-0.99, undefined`
	`-0.999, undefined`
	`-1.001, 2`
	`-1.01, 2`
	`-1.1, 2`

`> simpleLimitTable(g,1);`	`1.1, 2.01`
	`1.01, 2.0001`
	`1.001, 2.000001`
	`0.999, undefined`
	`0.99, undefined`
	`0.9, undefined`

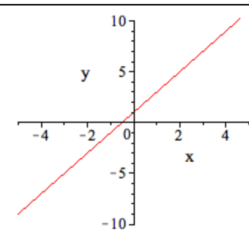
`> simpleLimitTable(g,4);`	`4.1, 3`
	`4.01, 3`
	`4.001, 3`
	`3.999, 10.994001`
	`3.99, 10.9401`
	`3.9, 10.41`

Figure 55. Group P_2's response to Lab 2, Problem 1 exploring mystery function g(x)

As with P_1, function m was a challenging exploration for the pair. When asked to plot the function, the pair correctly produced the apparently linear plot in Figure 56. They applied `simpleLimitTable` to the graph at x=2 but did not respond to the request to state the limiting value.

The pair was then given the task of refining the `simpleLimitTable` procedure by creating a new procedure, `simpleLimitTable2`, to approach a given point more closely. Specifically, the procedure evaluates the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point. The pair again produced an appropriate procedure shown in Figure 57 and utilized it to again examine the behavior of function m near x=2. This resulted in a table that was discrepant with the table produced by `simpleLimitTable`.

```
> plot ( m(x), x=-5..5, y=-10..10);
```




```
> simpleLimitTable(m,2);
    2.1, 5.2
    2.01, 5.02
    2.001, 5.002
    1.999, 4.998
    1.99, 4.98
    1.9, 4.8
```

Figure 56. Group P_2's exploration of mystery function m(x) near x=2 with a tolerance of 0.001

<table>
<tr><th>Definition of Procedure</th><th>Application of Procedure to m</th></tr>
<tr><td><pre>
> simpleLimitTable2 := proc(f, x)
   print (x+0.1, f(x+.1));
   print (x+.01, f(x+.01));
   print (x+.001, f(x+.001));
   print (x+.0001, f(x+.0001));
   print (x-0.1, f(x-.1));
   print (x-.01, f(x-.01));
   print (x-.001, f(x-.001));
   print (x-.0001, f(x-.0001));
end proc:
</pre></td><td><pre>
> simpleLimitTable2 (m,2);
      2.1, 5.2
      2.01, 5.02
      2.001, 5.002
      2.0001, 4.
      1.9, 4.8
      1.99, 4.98
      1.999, 4.998
      1.9999, 4.
</pre></td></tr>
</table>

Figure 57. Group P_2's implementation of `simpleLimitTable2`

Interestingly, the group changed the order of display for the evaluations performed by `simpleLimtTable2` potentially making the interpretation of results different from those produced by the first procedure, `simpleLimitTable`. Although, the procedure reflects an appropriate domain process, this change in ordering could potentially lead to confusion inferring limiting behavior.

`SimpleLimitTable2` was further refined, resulting in another new procedure, `simpleLimitTable3`, which approached the given point more closely. Specifically, the procedure evaluated the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point.

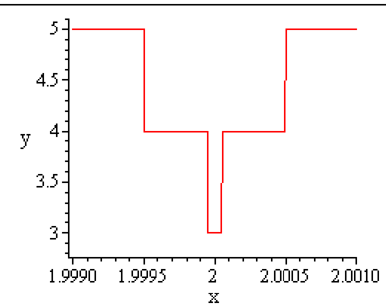
The pair again produced an appropriate procedure (see Figure 58), also with altered display order, and utilized it to again examine the behavior of function m near x=2. This resulted in a table that was discrepant with the table produced by `simpleLimitTable` and `simpleLimitTable2`. The group apparently only observed two of three discrepancies in result as they indicate "*the formula gave us 2 answers*" when asked to compare the results produced by the three implementations, `simpleLimitTable`, `simpleLimitTable2`, and `simpleLimitTable3`.

Definition of Procedure	**Application of Procedure to m**
`> simpleLimitTable3 := proc(f,x)`	`> simpleLimitTable3(m,2);`
`print (x+0.1, f(x+.1));`	`2.1, 5.2`
`print (x+.01, f(x+.01));`	`2.01, 5.02`
`print (x+.001, f(x+.001));`	`2.001, 5.002`
`print (x+.0001, f(x+.0001));`	`2.0001, 4.`
`print (x+.00001, f(x+.00001));`	`2.00001, 3.`
`print (x-0.1, f(x-.1));`	`1.9, 4.8`
`print (x-.01, f(x-.01));`	`1.99, 4.98`
`print (x-.001, f(x-.001));`	`1.999, 4.998`
`print (x-.0001, f(x-.0001));`	`1.9999, 4.`
`print (x-.0001, f(x-.00001));`	`1.9999, 3.`
`end proc:`	

Figure 58. Group P_2's implementation and demonstration of `simpleLimitTable3`

Finally the group was asked to look as closely as necessary at the graph of m so as to explain the contradictory results produced by `simpleLimitTable`, `simpleLimitTable2`, and `simpleLimitTable3`. The pair produced the plot and explanation in Figure 59. The graph clearly shows the unexpected behavior around x=2. From their response to problem eight, the pair infers that this implies there is no limit. The explanation given refers to the domain process and suggests some confusion of the domain and range processes.

```
> plot ( m(x), x=1.999..2.001, y=2.8..5.1);
```



8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

There is no limit, because X does not approach it

Figure 59. Group P_2's response to Lab 2, Problem 8 exploring mystery function m(x)

To summarize, the pair was able to construct the necessary procedures and demonstrated an understanding of appropriate domain processes attaining understandings outlined in APOS steps 1, 2, 3a but not 3b. The group, like programming group P_1, did not use the tool to explain any limiting behavior suggesting either a greater focus on the creation of the tools than on their use or an inability to construct a sequential argument relating to limiting behavior. Some confusion relating to understanding the coordination of the domain and range processes appears to be present. The change in the display order might also have added to this misunderstanding. This pair continues to experiment symbolically in the Maple CAS by varying parameter names. As a pair, the group continued to perform well utilizing the pair programming model again giving each other perfect peer evaluations.

Group P_3 (singleton)

The pair did not explore function f using evaluations as requested. Figure 60 shows the pairs implementation of the `simpleLimitTable` procedure and its usage. The procedure makes appropriate use of parameters and utilizes an appropriate domain process to function f at x=3.

Using this procedure, the students focused attention on understanding the behavior of functions g, h, k, and m. The pair appropriately applied the procedure and concluded that function g possessed discontinuities (see Figure 61), "*There is a discontinuity because there are undefined areas of the graph*" but failed to state specifically the location of these discontinuities. No discussion of limiting behavior is made.

Definition of Procedure	**Application of Procedure to f**

`> simpleLimitTable := proc( f, x )`
`print(x+0.1, f(x+.1) );`
`print(x+0.01, f(x+.01));`
`print(x+0.001, f(x+.001));`
`print(x-.1, f(x-.1));`
`print(x-.01, f(x-.01));`
`print(x-.001, f(x-.001));`
`end proc:`
	`> simpleLimitTable(f,3);`
	`3.1, 1.639344262`
	`3.01, 1.663893511`
	`3.001, 1.666388935`
	`2.9, 1.694915254`
	`2.99, 1.669449082`
	`2.999, 1.666944491`

Figure 60. Group P_3's implementation of `simpleLimitTable`

Problem #1 - Exploration of g(x) Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.	

`> simpleLimitTable(g,-1);`	`-0.9, undefined`
	`-0.99, undefined`
	`-0.999, undefined`
	`-1.1, 2`
	`-1.01, 2`
	`-1.001, 2`

`> simpleLimitTable(g,1);`	`1.1, 2.01`
	`1.01, 2.0001`
	`1.001, 2.000001`
	`0.9, undefined`
	`0.99, undefined`
	`0.999, undefined`

`> simpleLimitTable(g,4);`	`4.1, 3`
	`4.01, 3`
	`4.001, 3`
	`3.9, 10.41`
	`3.99, 10.9401`
	`3.999, 10.994001`

There is a discontinuity because there are undefined areas of the graph.	

Figure 61. Group P_3's response to Lab 2, Problem 1 exploring mystery function g(x)

For function h, the group correctly surmised the graph possessed a vertical asymptote in problem 2 but did not specify its location nor how they made this determination from the resulting limit tables (see Figure 62). They also mistakenly infer continuity as a graph with asymptotic behavior cannot be everywhere continuous.

For function k, the group mistakenly determines the graph possessed an asymptote but the output from `simpleLimitTable` does not support this conclusion (see Figure 63).

Problem #2 - Exploration of h(x) Explain the behavior of the graph h(x) at x= 0, and x= 3. Use your procedure simpleLimitTable as well as specific evaluations of h(x) to support your conclusions.	

`> simpleLimitTable(h,0);`	`0.1, 0.2`
	`0.01, 0.02`
	`0.001, 0.002`
	`-0.1, 0.01`
	`-0.01, 0.0001`
	`-0.001, 0.000001`

`> simpleLimitTable(h,3);`	`3.1, 10.`
	`3.01, 100.`
	`3.001, 1000.`
	`2.9, 5.8`
	`2.99, 5.98`
	`2.999, 5.998`

There is an asymptote in this graph; however, it is continuous everywhere but where there is an asymptote!	

Figure 62. Group P_3's response to Lab 2, Problem 2 exploring mystery function h(x)

Problem #3 - Exploration of k(x) Explain the behavior of the graph k(x) at x= 0, and x= 2. Use your procedure simpleLimitTable as well as specific evaluations of k(x) to support your conclusions.	

`> simpleLimitTable(k,0);`	`0.1, 1.1`
	`0.01, 1.01`
	`0.001, 1.001`
	`-0.1, 0.9`
	`-0.01, 0.99`
	`-0.001, 0.999`

`> simpleLimitTable(k,2);`	`2.1, 3.1`
	`2.01, 3.01`
	`2.001, 3.001`
	`1.9, 2.9`
	`1.99, 2.99`
	`1.999, 2.999`

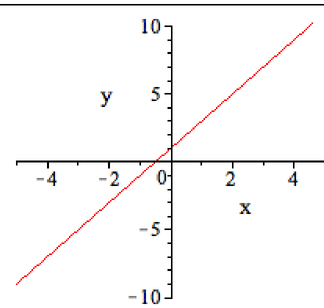
This graph also contains asymptotes creating discontinuity!	

Figure 63. Group P_3's response to Lab 2, Problem 2 exploring mystery function k(x)

In addressing the challenge problem, the pair correctly produced a plot that appears linear (see Figure 64). They further correctly applied `simpleLimitTable` to function m at x=2 but did not respond to the request to state the limiting value.

Interestingly, the group retyped the `simpleLimitTable` procedure with one modification; they changed the parameter originally called f with a new parameter named m (see Figure 65). Assumedly the pair believed the actual parameters' name must match the formal parameters' name. This is somewhat puzzling as when analyzing functions g, h, and k, such a modification was not deemed necessary. This brings to light some confusion with regard to the way parameters function in Maple.

```
> plot ( m(x), x=-5..5, y=-10..10);
```




```
> simpleLimitTable(m,2);
```

```
2.1, 5.2
2.01, 5.02
2.001, 5.002
1.9, 4.8
1.99, 4.98
1.999, 4.998
```

Figure 64. Group P_3's exploration of mystery function m(x) with tolerance 0.001

<table>
<tr><th>Before modification</th><th>After modification</th></tr>
<tr><td><pre>
> simpleLimitTable := proc(f, x)
      print(x+0.1, f(x+.1));
      print(x+0.01, f(x+.01));
      print(x+0.001, f(x+.001));
      print(x-.1,f(x-.1));
      print(x-.01, f(x-.01));
      print(x-.001, f(x-.001));
end proc:
</pre></td><td><pre>
> simpleLimitTable :=proc(m, x)
      print(x+0.1, m(x+.1));
      print(x+0.01, m(x+.01));
      print(x+0.001, m(x+.001));
      print(x-.1, m(x-.1));
      print(x-.01, m(x-.01));
      print(x-.001, m(x-.001));
end proc:
</pre></td></tr>
</table>

Figure 65. Group P_3's misunderstanding of formal versus actual parameters

Next, the pair then refined the `simpleLimitTable` procedure by creating a new procedure, `simpleLimitTable2`, that approached a given point x within 0.1, 0.01, 0.001, and 0.0001 of point x. The resulting procedure shown in Figure 66 was utilized to examine the behavior of function m near x=2 but due to a typographic error one of the function calls in the body of the procedure incorrectly referenced the original parameter name for the function f rather than m. This copy and paste error caused the generated table that would suggest an erroneous estimate of the limiting value. This suggested limit, while incorrect, still differed from the limit suggested by `simpleLimitTable` and, most importantly, was still overlooked by the pair; In fact, they claimed the results were identical.

<table>
<tr><th>Definition of Procedure</th><th>Application of Procedure to m</th></tr>
<tr><td><pre>
> simpleLimitTable2 := proc(m, x)
      print(x+0.1, m(x+.1));
      print(x+0.01, m(x+.01));
      print(x+0.001, m(x+.001));
      print(x+0.0001, f(x+.0001));
      print(x-.1, m(x-.1));
      print(x-.01, m(x-.01));
      print(x-.001, m(x-.001));
      print(x-0.0001, f(x-.0001));
end proc:
</pre></td><td><pre>
> simpleLimitTable2 (m,2);
  2.1, 5.2
  2.01, 5.02
  2.001, 5.002
  2.0001, 1.999960001
  1.9, 4.8
  1.99, 4.98
  1.999, 4.998
  1.9999, 2.000040001
</pre></td></tr>
</table>

Figure 66. Group P_3's implementation and demonstration `simpleLimitTable2`

`SimpleLimitTable2` was further refined, resulting in another new procedure, `simpleLimitTable3`, which approached the given point more closely. This procedure did not suffer from the same typographic error as `simpleLimitTable2` but instead had a new typographic error. Specifically, the evaluation at 0.0001 units from x is incorrect. See the bolded statement in Figure 67. Moreover, when they applied the procedure, they applied it to the wrong function. The group applied `simpleLimitTable3` to function f rather than m. The combination of this with the typographical error in `simpeLimitTable2` resulted in the students seeing the limiting value as the same as both procedures contained evaluations of f near 2. This might be a consequence of their misunderstanding and formal and actual parameters previously observed in Figure 65.

Definition of Procedure	**Application of Procedure to m**

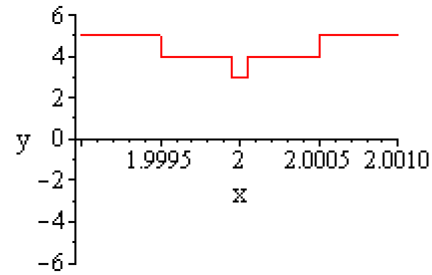
`> simpleLimitTable3 := proc( f, x )`
`print(x+0.1, f(x+.1) );`
`print(x+0.01, f(x+.01));`
`print(x+0.001, f(x+.001));`
`print(x+0.0001, f(x+.001))``;`
`print(x-.1, f(x-.1));`
`print(x-.01, f(x-.01));`
`print(x-.001, f(x-.001));`
`print(x-0.0001, f(x-.0001));`
`print(x-0.00001, f(x-.00001));`
`end proc:`
	`> simpleLimitTable3(`**`f`**`,2);`
	`2.1, 1.960784314`
	`2.01, 1.996007984`
	`2.001, 1.999600080`
	`2.0001, 1.999600080`
	`1.9, 2.040816327`
	`1.99, 2.004008016`
	`1.999, 2.000400080`
	`1.9999, 2.000040001`
	`1.99999, 2.000004000`
	
	NOTE: Function mistakenly applied to function f rather than m.

Figure 67. Group P_3's implementation and demonstration of `simpleLimitTable3`

Finally the group was asked to look as closely as necessary at the graph of m so as to explain the contradictory results produced by `simpleLimitTable`, `simpleLimitTable2`, and `simpleLimitTable3`. The pair produced the plot and explanation in Figure 68 that clearly showed the unexpected behavior around x=2. From their response to Problem 8, the pair infers this implied a limit but one that is dependent upon the continuity of the function and its value at the limiting point; they suggest zooming into the graph would help one determine the limit but that plugging the x value into the "equation" would be necessary to verify the limit.

To summarize, the pair was able to construct the necessary procedures and demonstrate an understanding of appropriate domain processes attaining understandings outlined in APOS steps 1, 2, 3a, but not 3b. The group did not use the tool to explain any limiting behavior. As with the other programming groups, more attention seems directed toward the domain process. Additionally, several typographical errors lead to serious misinterpretations. Confusion related to actual and formal parameters to procedures in Maple also lead to erroneous data using `simpleLimitTable2` and `simpleLimitTable3`. As a pair, the group continued to perform well utilizing the pair programming model again giving each other perfect peer evaluations.

```
> plot ( m(x), x=1.999..2.001, y=-6..6);
```



8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

You would have to zoom all the way in to the ditch at two to make sure there is continuity there. Plug it into the equation to verify the limit found is correct!

Figure 68. Group P_3's response to Lab 2, Problem 8 exploring mystery function m(x)

Group P_4 (singleton)

In exploring function f near x=1, 2, and -3, the pair performed a few evaluations on one side of the points of interest as shown in Figure 69 . This was the only programming pair to make any sort of sequential argument in response to this exploration. Of note is that the arguments are one-sided, the pair only considers limiting behavior from one side.

The pair produced the `simpleLimitTable` procedure, shown in Figure 70, that approached the given domain value even more closely than the requested to a tolerance of 0.001 from below; Notice, on the highlighted line, the procedure evaluates the function to within 0.0001 on the left of point x. The pair subsequently demonstrated appropriate use of parameters, an understanding of the relevant domain process, as well as proper use of the parameters and functions.

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."

```
> f (1); f(1.1); f(1.01);
                                       2.500000000
                                       2.439024390
                                       2.493765586
```

Description of behavior: As "x" nears 1, "f" gets closer to 2.5

2) Explore and Describe the behavior of function f(x) at and near x=2.

```
> f(2); f(2.1); f(2.01);
                                        undefined
                                       1.960784314
                                       1.996007984
```

Description of behavior: At 2, "f" is undefined, but the limit as "x" approaches 2 is 2.

3) Explore and Describe the behavior of function f(x) at and near x= -3.

```
> f(-3); f(-2.9); f(-3.1);
Error, (in MysteryFunctions:-f) numeric exception: division by zero
                                          100.
                                          -100.
```

Description of behavior: The function "f" is not defined at -3.

Figure 69. Group P_4's response to Lab 2, Problems 1-3 exploring mystery function f(x)

Definition of Procedure

```
> simpleLimitTable := proc( f, x )
  print( x+0.1, f(x+.1));
>   print( x+0.01, f(x+.01));
>   print( x+.001, f(x+.001));
>   print( x-.0001, f(x-.0001));
>   print( x-.001, f(x-.001));
>   print( x-.01, f(x-.01));
>   print( x-.1, f(x-.1));
> end proc:
```

Figure 70. Group P_4's implementation of `simpleLimitTable`

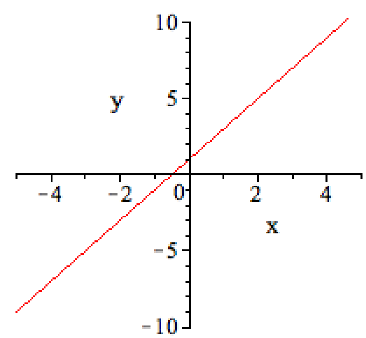
Using this procedure, the students focused their attention on understanding the behavior of functions g, h, k, and m. For functions g, h, and k, the pair appropriately applied their procedure but failed to state any conclusions about the functions. Only exploration of function g is shown in Figure 71; Results for functions h and k were similar. As with other programming groups, the lack of discussion relating to limiting behavior suggested greater focus on the development of the procedure than on its subsequent application; the tables merely demonstrate the procedure is operating as expected.

The group correctly produced a plot of mystery function m which appears linear in Figure 72. Then they applied `simpleLimitTable` to the function at x=2 but again did not respond to the request to state the limiting value.

Problem #1 - Exploration of g(x) Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.	
`> simpleLimitTable(g,-1);`	-0.9, undefined -0.99, undefined -0.999, undefined -1.0001, 2 -1.001, 2 -1.01, 2 -1.1, 2
`> simpleLimitTable(g,1);`	1.1, 2.01 1.01, 2.0001 1.001, 2.000001 0.9999, undefined 0.999, undefined 0.99, undefined 0.9, undefined
`> simpleLimitTable(g,4);`	4.1, 3 4.01, 3 4.001, 3 3.9999, 10.99940001 3.999, 10.994001 3.99, 10.9401 3.9, 10.41

Figure 71. Group P_4's response to Lab 2, Problem 1 exploring mystery function g(x)

```
> plot ( m(x), x=-5..5, y=-10..10);
```



`> simpleLimitTable(m,2);`	2.1, 5.2 2.01, 5.02 2.001, 5.002 1.9999, 4 1.999, 4.998 1.99, 4.98 1.9, 4.8

Figure 72. Group P_4's exploration of mystery function m(x) near x=2 with tolerance 0.001

The pair was then given the task of refining the `simpleLimitTable` procedure by creating a new procedure, `simpleLimitTable2` that approached the given point more closely. Specifically, the procedure evaluated the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point. The pair again produced an appropriate procedure and utilized it to examine the behavior of function m near x=2 (see Figure 73). This was intended to produce output that was clearly discrepant with the table produced by `simpleLimitTable` but since the group used a smaller tolerance than requested in `simpleLimitTable`, the results do not contradict one another in the manner intended. However, the group did make the important observation that a small change in the input to function m resulted in a rather large change in the value produced by the function. "*The value 0.0001 makes a big difference in the function output.*" suggesting they noticed a discrepancy in the two tables.

Definition of Procedure	**Application of Procedure to m**

`> simpleLimitTable2 := proc( f, x )`
`print(x+0.1,f(x+.1) );`
`print(x+0.01,f(x+.01) );`
`print(x+.001,f(x+.001) );`
`print(x+.0001,f(x+.0001) );`
`print(x-0.0001,f(x-0.0001) );`
`print(x-.001,f(x-.001) );`
`print(x-.01,f(x-.01) );`
`print(x-.1,f(x-.1) );`
`end proc:`
	`> simpleLimitTable2 (m,2);`
	`2.1, 5.2`
	`2.01, 5.02`
	`2.001, 5.002`
	`2.0001, 4.`
	`1.9999, 4.`
	`1.999, 4.998`
	`1.99, 4.98`
	`1.9, 4.8`

Figure 73. Group P_4's implementation of demonstration of `simpleLimitTable2` on mystery function m(x) near x=2 with tolerance 0.0001

SimpleLimitTable2 was further refined, resulting in another new procedure, simpleLimitTable3, which approached the given point more closely. Specifically, the procedure it evaluated the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point (see Figure 74).

The pair again produced an appropriate procedure and utilized it to again examine the behavior of function m near x=2. This resulted in a table that was discrepant with the table produced by simpleLimitTable and simpleLimitTable2.

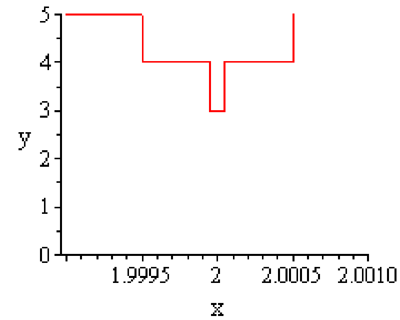
Finally the group was asked to look as closely as necessary at the graph of m so as to explain the dissimilar results produced by simpleLimitTable, simpleLimitTable2, and simpleLimitTable3. The pair produced the plot and explanation shown in Figure 75. The graph more clearly showed the unexpected behavior around x=2. From their response to problem eight, the pair has a very clear understanding that this zooming process must continue in order to determine the limiting value.

Definition of Procedure	Application of Procedure to m

`> simpleLimitTable3 := proc( f, x )`
`print(x+0.1,f(x+.1) );`
`print(x+0.01,f(x+.01) );`
`print(x+.001,f(x+.001) );`
`print(x+.0001,f(x+.0001) );`
`print(x+.00001,f(x+.00001) );`
`print(x-.00001,f(x-.00001) );`
`print(x-.0001,f(x-.0001) );`
`print(x-0.001,f(x-0.001) );`
`print(x-.01,f(x-.01) );`
`print(x-.1,f(x-.1) );`
`end proc:`
	`> simpleLimitTable3(m,2);`
	`2.1, 5.2`
	`2.01, 5.02`
	`2.001, 5.002`
	`2.0001, 4.`
	`2.00001, 3.`
	`1.99999, 3.`
	`1.9999, 4.`
	`1.999, 4.998`
	`1.99, 4.98`

Figure 74. Group P_4's implementation of demonstration of simpleLimitTable3 on mystery function m(x) near x=2 with tolerance 0.00001

```
> plot ( m(x), x=1.999..2.001, y=0..5.);
```



8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

Decrease the window of "X" over and over again (by a hundredth, thousandth, millionth, etc.) until the limit stays the same, and you can be convinced the limit is the number you keep getting back.

Figure 75. Group P_4's response to Lab 2, Problem 8 exploring mystery function m(x)

To summarize, this group pair was able to construct the necessary procedures and demonstrated an understanding of appropriate domain processes attaining understandings outlined in APOS steps 1, 2, 3a and 3b. The group used to tool to describe limiting behavior of function f with sequential arguments. This was the only programming group to make such justifications. Moreover, in their analysis of mystery function m, the group demonstrates a very clear understanding of both the domain and range processes as well as their respective coordination in the limiting process. As this was a singleton group, no comment regarding peer interaction is made.

Group N_1

As shown in Figure 76, the group consisting of two lower-performing females was able to perform function evaluation of function f but was unable to construct a sequential argument suggesting limiting behavior. As can be seen in the following, in their analysis of function f, they rely strictly on evaluation for determining limiting values.

For the non-programming groups, the `simpleLimitTable` procedure operated in a slightly different fashion; it took three parameters, a function, a point, and a power of 10 tolerance. For example, a call of the form `simpleLimitTable( g, -1, 0.001)` would result in a table of function evaluations of g(x) near x=-1 to a tolerance of 0.001. When asked to use this `simpleLimitTable` procedure to explore the behavior of functions g, h, and k at specific points, the pair was unable to use the tool due to a lack of understanding of parameters to a procedure.

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."

```
> f(1);
                              2.500000000
```

As x approaches to one from both sides, the limit of the function is 2.5

2) Explore and Describe the behavior of function f(x) at and near x=2.

```
> f(2);
                               Undefined
```

As x approaches near two from both sides, the limit of function is undefined

3) Explore and Describe the behavior of function f(x) at and near x=-3.

```
> f(3);
                              1.666666667
```

As x approaches near two from both sides, the limit of function is 1.67

Figure 76. Group N_1's response to Lab 2, Problems 1-3 exploring mystery function f(x)

The pair clearly assumed that the parameters following the function name indicated the points at which to explore the limiting behavior. This is not unreasonable, yet they did not seem concerned with the lack of resulting output produced by their actions (see Figure 77). An understanding of parameters to procedures was lacking; they failed to understand the procedure always had three parameters and that the third parameter represented a tolerance to which to approach the given point. Of note is that one of the pair was concurrently enrolled in a computer science course taught by the instructor in which the development of such understanding is specifically addressed; however, this student struggled with many computer science concepts in this other course. In each case, the tool produced no output, yet the students drew conclusions; no explanation is given as to how these conclusions were derived.

Problem #1 - Exploration of g(x)
Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.

```
> simpleLimitTable( g, -1, 1, 4);
```

As x approaches -1,1,4, the value of the function increases; There is a discontinuity at the x value 1

Problem #2 - Exploration of h(x)
Explain the behavior of the graph h(x) at x= 0, and x= 3. Use your procedure simpleLimitTable as well as specific evaluations of h(x) to support your conclusions.

```
> simpleLimitTable( h, 0, 3);
```

As x approaches 0 and 3 from both sides, the value of the function is 0 and 6

Problem #3 - Exploration of k(x)
Explain the behavior of the graph k(x) at x= 0, and x= 2. Use your procedure simpleLimitTable as well as specific evaluations of k(x) to support your conclusions.

```
> simpleLimitTable( k, 0, 2);
```

as x approaches 0 from the left, the value of the function is 1, and there is a discontinuity as k approaches 2.

Figure 77. Group N_1's response to Lab 2, Problems 1-3 exploring mystery functions g(x), h(x), and k(x)

After having the procedure parameters clarified by the instructor, the pair was able to explore function m. On problem 2, they examined the functions value at x=3 and produced a limit table to within 0.001 of x=2 (see Figure 78). They made no comment regarding the discrepancy between the trend in the table and the functions value. Interestingly, the pair indicated that there could be more than one limiting value at x=2 as each of the limit tables of differing tolerances suggested a different limiting value.

2. Using simpleLimitTable, estimate the limit as x approaches 2.

```
> m(2);
                                    3.
> simpleLimitTable( m,2,0.001);
       1.900000        4.800000
       1.990000        4.980000
       1.999000        4.998000
       2.001000        5.002000
       2.010000        5.020000
       2.100000        5.200000
```

3. Use the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point.

```
> simpleLimitTable( m,2,0.0001);
       1.900000        4.800000
       1.990000        4.980000
       1.999000        4.998000
       1.999900        4.000000
       2.000100        4.000000
       2.001000        5.002000
       2.010000        5.020000
       2.100000        5.200000
```

4. Do you notice anything when you compare the results of step 2 and step 3?
yes because, the limit at the point closer to 2, is 5, but the function of the numbers closer to 2 is 4.

5. Using the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point.

```
> simpleLimitTable(m,2,0.00001);
       1.900000        4.800000
       1.990000        4.980000
       1.999000        4.998000
       1.999900        4.000000
       1.999990        3.000000
       2.000010        3.000000
       2.000100        4.000000
       2.001000        5.002000
       2.010000        5.020000
       2.100000        5.200000
```

6. Do you notice anything when you compare the results of steps 2, 3, and 4?
They have different limits

Figure 78. Group N_1's response to Lab 2, Problems 2-6 exploring mystery function m(x) near x=2

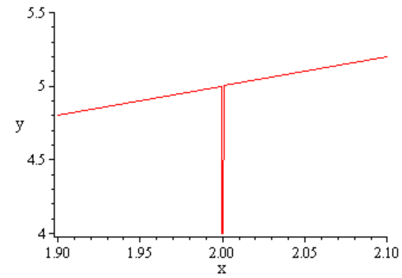
Finally the group was asked to look closely at the graph of m so as to explain the dissimilar results produced by `simpleLimitTable`, `simpleLimitTable2`, and `simpleLimitTable3`. The pair produced the plot and explanation shown in Figure 79. The graph more clearly showed the unexpected behavior around x=2. From their response to Problem 8, the pair suggested one requires a graph and a table to support a particular limiting value.

To summarize, the pair initially had difficulty making inferences using a sequential argument. In fact, they did not see necessity for information beyond a functions value at the limiting point. They initially had difficulty using the `simpleLimitTable` tool to explore a functions behavior around a given point. After discussion with the instructor, these difficulties were addressed and the pair made relevant observations relating to the challenge function, m.

The group made a significant stride forward in understanding in this lab as they came to understand sequential inference in the context of limits and the necessity of understanding a functions behavior around, rather than at, the limiting point. By the end of the lab, the pair demonstrated an understanding of appropriate domain processes attaining understandings outlined in APOS steps 1, 2, 3a, and 3b. This group gave each other perfect peer reviews.

<table>
<tr><td>

```
> plot ( m(x), x=1.9..2.1, y=4..5.5);
```



</td></tr>
<tr><td>8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

a simple limit table and a plot graph, because you can see when the points get closer and closer to a certain value.</td></tr>
</table>

Figure 79. Group N_1's response to Lab 2, Problem 8 exploring mystery function m(x) near x=2

Group N_2

When exploring function f near x=1, see Figure 80, the group evaluated the function at x=1 and explored the trend in function values as x approached one from the left (one-sided limit). This was done again at x=2. At the point x=-3, the approach was from the right rather than from the left assumedly due to a misunderstanding of the order properties of the real number line. This process of exploration suggests that one must approach the limiting value from either (but not both) sides of the limiting point. No statement relating to specific limits was given.

The group was able to subsequently utilize the `simpleLimitTable` tool in exploring functions g, h, and k. When exploring the function g near x=-1, x=1, and x=4, the pair produced a sequence of evaluations manually as well as using the `simpleLimitTool`. For brevity, only the evaluations of g near x=1 are shown in Figure 81. The students were able to productively discern relevant behavior at the given points. Interestingly, in spite of the fact that `simpleLimitTable` produced results identical to the results of their individual evaluations, they continued to perform evaluations in addition to using the tool in subsequent problems suggesting a lack of comfort with the tool- a hesitancy to adopt the new tool.

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."

```
> f(.9);
> f(.99);
                                 2.564102564
                                 2.506265664
> f(.999);
                                 2.500625156
> f(1);
                                 2.500000000
```

2) Explore and Describe the behavior of function f(x) at and near x=2.

```
> f(1.9);
                                 2.040816327
> f(1.99999999);
                                 2.000000004
> f(2);
                                  undefined
```

3) Explore and Describe the behavior of function f(x) at and near x=-3.

```
> f(-2.9);
                                    100.
> f(-2.99);
                                    1000.
> f(-3);
Error, (in MysteryFunctions:-f) numeric exception: division by zero
```

Figure 80. Group N_2 's response to Lab 2, Problems 1-3 exploring mystery function f(x)

Problem #1 - Exploration of g(x)
Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.

```
>g(.9);g(.99);g(.999);g(.9999);g(1);g(1.0001);g(1.001);g(1.01);g(1.1)
;
simpleLimitTable(g,1,.0001);
       undefined
       undefined
       undefined
       undefined
       undefined
       2.00000001
       2.000001
       2.0001
       2.01

       0.900000            NaN
       0.990000            NaN
       0.999000            NaN
       0.999900            NaN
       1.000100       2.000000
       1.001000       2.000001
       1.010000       2.000100
       1.100000       2.010000
```

Figure 81. Group N_2 's response to Lab 2, Problem 1 exploring mystery function g(x) near x=1

The conclusions, derived from explorations of functions g and h, are shown in Figure 82. As can be seen, the students very accurately described the functions behavior based upon their use of the tool. A graph of the mystery function is shown in Figure 82 for reference; the pair did not have access to a graphical representation for this mystery function. Function k was similarly analyzed with appropriate tables and evaluations produced without drawing any conclusion regarding the behavior at x=0 and x=2.

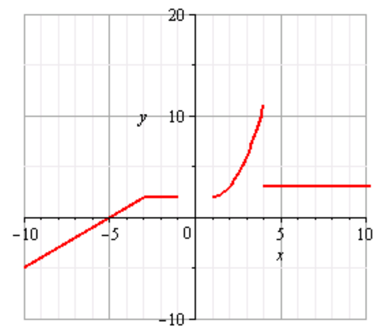
<table>
<tr><th>Conclusions related to Function g</th><th>Actual graph of g</th></tr>
<tr><td>

as x->-1 from the right, it does not exist. However, as x->-1 from the left, it approaches 2.

as x->1 from the left, it doesn't exist, but as it approaches from the right, y approaches 2.

As x->4 from the left, the limit is 11, but it jumps to a horizontal line at y=3 from the right. This all means that there is somethin' funky in the interval (-1,1) and a discontinuity at x=4

</td><td>

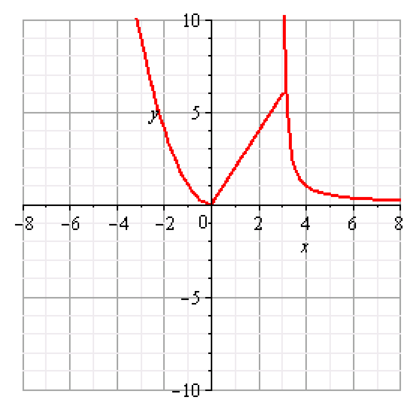


Jump at x=4, no left limit at x=1, no right limit at x=-1.

</td></tr>
<tr><th>Conclusions related to Function h</th><th>Actual graph of h</th></tr>
<tr><td>

As x->0 from the left and right, the limit is 0. As x->3 from the left, the limit is 6, but as it approaches 3 from the right, it goes to infinity and beyond, implying a VA there.

</td><td>



Corner at x=0 and a Vertical asymptote at x=3.

</td></tr>
</table>

Figure 82. Group N_2 's conclusions relating to mystery functions g(x) and h(x)

On challenge function m, the pair deduced a limit as x approached 2 of 5 using a tolerance of 0.001. Notably, the students did not note differences in the implied limiting values as smaller tolerances were used, despite being asked the question twice (see Figure 83). Upon zooming in on the graph at x=2 (as instructed), they were able to produce a graph demonstrating the true local behavior near x=2 but did not comment on the prior discrepant data.

2. Using <u>simpleLimitTable</u>, estimate the limit as x approaches 2.

```
> simpleLimitTable(m,2,.001);
        1.900000        4.800000
        1.990000        4.980000
        1.999000        4.998000
        2.001000        5.002000
        2.010000        5.020000
        2.100000        5.200000
```

looks like 5 to us.

3. Use the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point**.**

```
> simpleLimitTable(m,2,.0001);
        1.900000        4.800000
        1.990000        4.980000
        1.999000        4.998000
        1.999900        4.000000
        2.000100        4.000000
        2.001000        5.002000
        2.010000        5.020000
        2.100000        5.200000
> simpleLimitTable(m,2,.001);
        1.900000        4.800000
        1.990000        4.980000
        1.999000        4.998000
        2.001000        5.002000
        2.010000        5.020000
        2.100000        5.200000
> simpleLimitTable(m,2,.01);
        1.900000        4.800000
        1.990000        4.980000
        2.010000        5.020000
        2.100000        5.200000
        1.900000        4.800000
        2.100000        5.200000
```

4. Do you notice anything when you compare the results of step 2 and step 3?

No not really.

5. Using the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point.

```
> simpleLimitTable(m,2,.1);
        1.900000        4.800000
        2.100000        5.200000
> simpleLimitTable(m,2,.01);
        1.900000        4.800000
        1.990000        4.980000
        2.010000        5.020000
        2.100000        5.200000
> simpleLimitTable(m,2,.001);
        1.900000        4.800000
        1.990000        4.980000
        1.999000        4.998000
        2.001000        5.002000
        2.010000        5.020000
        2.100000        5.200000
> simpleLimitTable(m,2,.0001);
        1.900000        4.800000
        1.990000        4.980000
        1.999000        4.998000
        1.999900        4.000000
        2.000100        4.000000
        2.001000        5.002000
        2.010000        5.020000
        2.100000        5.200000
> simpleLimitTable(m,2,.00001);
        1.900000        4.800000
        1.990000        4.980000
        1.999000        4.998000
        1.999900        4.000000
        1.999990        3.000000
        2.000010        3.000000
        2.000100        4.000000
        2.001000        5.002000
        2.010000        5.020000
        2.100000        5.200000
```

6. Do you notice anything when you compare the results of steps 2, 3, and 5?

It just keeps getting closer to 5.

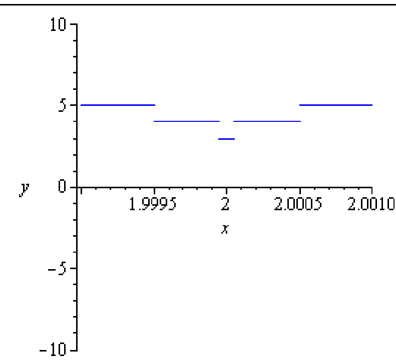
Figure 83. Group N_2's response to Lab 2, Problems 2-6 exploring mystery function m(x) near x=2

Finally the group looked closely at the graph of m and explained dissimilar results produced by `simpleLimitTable`, `simpleLimitTable2`, and `simpleLimitTable3`. The pair produced the plot and explanation shown in Figure 84. The graph clearly shows somewhat erratic behavior around x=2. In spite of their ability to make inferences from previous sequential evaluations, their response to Problem 8 suggests some definite formula exists yielding the limiting value, i.e. a limit is an object.

This group made effective use of the `simpleLimitTable` procedure to analyze function behavior. Their evaluations and their respective order indicate an understanding of the domain process that appears more refined than their initial understanding. Initially, the pair determined limits using one-sided sequential arguments and function evaluations. Later, they used two-sided arguments as evidenced by subsequent descriptions of function behavior for functions g and h. Understanding of APOS Steps 1, 2, 3a, and 3b is clearly demonstrated.

7. Look more closely at the graph around the point x=2 and explain the results found in steps 1-6.

```
> plot( m(x),x=1.999..2.001, y=-10..10,
color=blue, discont=true);
```



8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

The approval of the instructor *and/or entering in the correct formula for the limit in Maple (which we may or may not know).*

Figure 84. Group N_2 's response to Lab 2, Problems 7-8, Exploring mystery function m(x) near x=2

With regard to their group dynamics, this pair consisted unintentionally of two very close friends. As can be seen in their response to Problem 8, the pair often offered humorous responses to questions – occasionally at the expense of honestly answering the posed question. More significantly, however, their interaction during the lab frequently violated the rules of interaction defined by the pair-programming paradigm. Specifically, the two would switch roles more frequently than as outlined- much more frequently. This tended to happen seamlessly due to their close personal relationship. This more frequent changing of roles had the effect of blurring much of the distinction between the driver and navigator with both partners effectively performing both roles nearly simultaneously. Once noted, the instructor made a conscientious effort to be aware of the frequency of their role changes.

Group N_3

In exploring function f near x=1, 2, and -3, the group performed a few relevant evaluations on both sides of the respective points and provided accurate descriptions of limiting behavior near x=1, and x=2 (See Figure 85). Near x=1, the pair evaluated the function at the point x=1 and at points near x=1 on both sides correctly inferring the limiting value at x=1. At the point x=-3, they evaluated only at a single point on each side of x=-3 yet correctly deduce the functions behavior around x=-3.

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."
2) Explore and Describe the behavior of function f(x) at and near x=2.
3) Explore and Describe the behavior of function f(x) at and near x= -3.

```
> f(1);f(1);f(2);f(3);f(1.99999);
f(2.0000001);f(1.01);f(1.0000001);f(0.999999999);
                                 2.500000000
                                     5.
                                  undefined
                                 1.666666667
                                 2.000004000
                                 1.999999960
                                 2.493765586
                                 2.499999938
                                 2.500000001
> f(-2.9999999);f(-3.0000001);
                                         8
                                   1.0 10
                                         8
                                  -1.0 10
```

ANSWERS:
1) As you approach 1 from both the left and right, the value approaches 2.5.
2) As you approach 2 from both the left and right, the value approaches 2.
3) As x approaches 3 from the left it approaches negative infinity, and from the right it approaches infinity.

Figure 85. Group N_3's response to Lab 2, Problems 1-3, Exploring mystery function f(x)

For function g, the pair appropriately applied `simpleLimitTable` (See Figure 86). The pair noted that it appeared the slope of the graph of g was zero the left of x=-1. They claimed that there was no point at x=-1 but did not justify this with an appropriate evaluation. They erroneously inferred a zero slope to the right of x=1 and the left of x=4. No discussion of limiting behavior was offered. Their response is consistent with their initial limit conception that a limit "*describes how a function moves as x moves towards a certain point.*"

Problem #1 - Exploration of g(x) Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.		
`> simpleLimitTable(g,-1,0.001);`	`-1.100000`	`2.000000`
	`-1.010000`	`2.000000`
CONCLUSION:	`-1.001000`	`2.000000`
	`-.999000`	`NaN`
As the graph goes to -1 from the left it has a slope of zero, when it hits -1 it doesn't exist.	`-.990000`	`NaN`
	`-.900000`	`NaN`
`> simpleLimitTable(g,1,0.001);`	`.900000`	`NaN`
	`.990000`	`NaN`
CONCLUSION:	`.999000`	`NaN`
	`1.001000`	`2.000001`
As the graph goes to 1 from the right it has a slope of zero and when it hits 1 it doesn't exist after.	`1.010000`	`2.000100`
	`1.100000`	`2.010000`
`> simpleLimitTable(g,4,0.001);`	`3.900000`	`10.410000`
	`3.990000`	`10.940100`
CONCLUSION:	`3.999000`	`10.994001`
	`4.001000`	`3.000000`
As the graph goes to 4 from the left its slope is zero as it comes from the right the slope is zero but at different y values.	`4.010000`	`3.000000`
	`4.100000`	`3.000000`

Figure 86. Group N_3's response to Lab 2, Problem 1, Exploring mystery function g(x)

With function h, the pair did begin to offer descriptions of limiting behavior. The pair correctly deduced a discontinuity at x=3 and a limit of 0 at x=0 (see Figure 87). They failed classify the discontinuity at x=3 correctly, however. For function k, see Figure 88, a correct limit was deduced at x=1 and at x=2, however, they did not check to see if there was a point at either location. Thus they did not infer the presence of a hole at x=2.

Problem #2 - Exploration of h(x)

Explain the behavior of the graph h(x) at x= 0, and x= 3. Use your procedure simpleLimitTable as well as specific evaluations of h(x) to support your conclusions.

`> simpleLimitTable( h, 0, .0001);`	-.100000	.010000
	-.010000	.000100
the graph h(x) at 0 the limit is 0	-.001000	.000001
	-.000100	.000000
	.000100	.000200
	.001000	.002000
	.010000	.020000
	.100000	.200000
`> simpleLimitTable( h, 3, .001);`	2.900000	5.800000
	2.990000	5.980000
the graph of h(x) has a jump discontinuity	2.999000	5.998000
	3.001000	1000.000000
	3.010000	100.000000
	3.100000	10.000000

Figure 87. Group N_3's response to Lab 2, Problem 2, Exploring mystery function h(x)

Problem #3 - Exploration of k(x)

Explain the behavior of the graph k(x) at x= 0, and x= 2. Use your procedure simpleLimitTable as well as specific evaluations of k(x) to support your conclusions.

`> simpleLimitTable(k, 0, .001);`	-.100000	.900000
	-.010000	.990000
The limit of k(x) is 1 at the point 0.	-.001000	.999000
	.001000	1.001000
	.010000	1.010000
	.100000	1.100000
`> simpleLimitTable(k, 2, .001);`	1.900000	2.900000
	1.990000	2.990000
The limit for the value x=2 of the function k(x) is 3.	1.999000	2.999000
	2.001000	3.001000
	2.010000	3.010000
	2.100000	3.100000

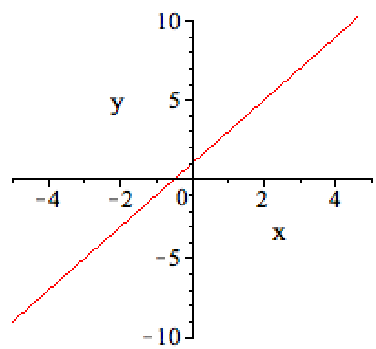
Figure 88. Group N_3's response to Lab 2, Problem 3, Exploring mystery function k(x)

Function m demonstrated the pairs clear understanding of the limiting process. When asked to plot the function, the pair correctly produced a plot that appears linear (see Figure 89). Using the graph, they made a reasonable inference regarding the limit as x approaches 2. They subsequently utilized `simpleLimitTable` to the graph at x=2 to support their claim.

In Figure 90, the pair was then given the task of refining the closeness to x=2 by decreasing the step-size. Specifically, the students evaluated the function at points within 0.01, 0.001, 0.0001, and 0.00001 of the given point. The discrepancy in limiting value was noted in problem four but not five.

```
> plot ( m(x), x=-5..5, y=-10..10);
```

The limit at x=2 is 5.




```
> simpleLimitTable(m,2,0.001);
```

The limit as x approached two is 5.

1.900000	4.800000
1.990000	4.980000
1.999000	4.998000
2.001000	5.002000
2.010000	5.020000
2.100000	5.200000

Figure 89. Group N_3's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerance 0.001

3. Use the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point.

```
> simpleLimitTable(m,2,.0001);
       1.900000        4.800000
       1.990000        4.980000
       1.999000        4.998000
       1.999900        4.000000
       2.000100        4.000000
       2.001000        5.002000
       2.010000        5.020000
       2.100000        5.200000
> simpleLimitTable(m,2,.001);
       1.900000        4.800000
       1.990000        4.980000
       1.999000        4.998000
       2.001000        5.002000
       2.010000        5.020000
       2.100000        5.200000
> simpleLimitTable(m,2,.01);
       1.900000        4.800000
       1.990000        4.980000
       2.010000        5.020000
       2.100000        5.200000
```

4. Do you notice anything when you compare the results of step 2 and step 3?

EXPLAIN HERE.

As you get closer to the .0001 increment, it shows the limit as being 4 instead of 5.

5. Using the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point.

```
> simpleLimitTable(m,2,.1);
       1.900000        4.800000
       2.100000        5.200000
```

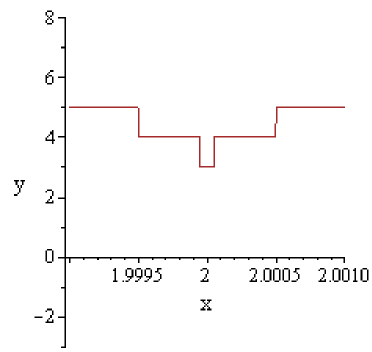
applications at tolerances of 0.01, 0.001, and 0.0001 not shown

```
> simpleLimitTable(m,2,.00001);
       1.900000        4.800000
       1.990000        4.980000
       1.999000        4.998000
       1.999900        4.000000
       1.999990        3.000000
       2.000010        3.000000
       2.000100        4.000000
       2.001000        5.002000
       2.010000        5.020000
       2.100000        5.200000
```

Figure 90. Group N_3's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerances 0.1, 0.01, 0.001, 0.0001 and 0.00001

Finally the group was asked to look as closely as necessary at the graph of m so as to explain the contradictory results. The pair produced the plot and explanation shown in Figure 91. The graph more clearly showed the unexpected behavior around x=2. From their response to problem eight, the pair clearly understands the challenge-response nature of the limiting procedure.

```
> plot( m(x), x= 1.999..2.001, y= -3..8);
```



8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

Getting infinitely close to the 2 could produce the correct limit value.

Figure 91. Group N_3's response to Lab 2, Problem 8, Exploring mystery function m(x) near x=2

For function m, they correctly deduced a limiting value of 5 at x=2 with a tolerance of 0.001. They noted that upon decreasing the tolerance, a different limiting value is found. They described the graph as having a blemish or imperfection as a consequence. They suggested one would need to get arbitrarily close to x=2 to determine the true limiting value.

To summarize, the pair demonstrated an understanding of appropriate domain and range processes attaining understandings outlined in APOS steps 1, 2, 3a, and 3b. As a pair, the group performed well utilizing the pair programming model again giving each other perfect peer evaluations. Like group N_2, both students were very engaged in the explorations, consistently aware of time spent by each member in their respective roles, and keenly aware of what their partner was doing. They were always eager to switch roles. I attempted to make them aware of specific duties prescribed by the respective navigator and driver roles within the pair programming model and have them focus on their individual roles rather than having this distinction blurred by their close interactions.

Group N_4

When exploring the behavior of f near x=1,2, and 3, the pair explored both sides but did not evaluate the function at the points, and did not state the limiting value they determined (see Figure 92). Their exploration around these points did not represent a careful sequential exploration but rather a single evaluation on each side of the points in question.

When exploring function g, shown in Figure 93, the pair argued there were vertical asymptotes at x=1, -1, and 4 and horizontal asymptotes at y=2 and y=3. Using local behavior, they made statements relating to global behavior rather than the local behavior requested. They did not correctly interpret the output from the tool.

In the lab, you will explore the behavior of an unknown function f(x). You are to consider the functions behavior at the points x = 1, 2, and -3. The only permitted action involving the function is evaluation, i.e. to evaluate a function f(x) at x=10;

```
> f(10);
                                    0.7692307692
```

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."

```
> f(1.5);
                                     2.222222222
> f(1.1);
                                     2.439024390
> f(0.99);
                                     2.506265664
```

2) Explore and Describe the behavior of function f(x) at and near x=2.

```
> f(2.5);
                                     1.818181818
> f(2.1);
                                     1.960784314
> f(1.99);
                                     2.004008016
```

3) Explore and Describe the behavior of function f(x) at and near x=-3.

```
> f(-3.1);
                                        -100.
> f(-2.9);
                                        100.
> f(-3.0001);
                                             5
                                      -1.0 10
```

Figure 92. Group N_4's analysis of mystery function f(x)

Problem #1 - Exploration of g(x)
Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.

```
> simpleLimitTable( g,-1,0.01);
      -1.100000        2.000000
      -1.010000        2.000000
       -.990000             NaN
       -.900000             NaN
> simpleLimitTable(g,1,.01);
        .900000             NaN
        .990000             NaN
       1.010000        2.000100
       1.100000        2.010000
> simpleLimitTable(g,4,.01);
       3.900000       10.410000
       3.990000       10.940100
       4.010000        3.000000
       4.100000        3.000000
```

Vertical asymptote at x=1 and -1 and 4; Horizontal asymptote at y=2 and 3, at least we think so...

Figure 93. Group N_4's analysis of mystery function g(x)

For function h (see Figure 94), they incorrectly argue that there is a vertical asymptote at x=0 perhaps because both values on either side of x=0 are positive. Correctly inferred a vertical asymptote at x=3. Similarly for function k shown in Figure 95, the pair also erroneously argues that there are vertical asymptotes at x=0 and x=2. Clearly, the group does not know how to interpret the output from the `simpleLimitTable` tool.

Problem #2 - Exploration of h(x)
Explain the behavior of the graph h(x) at x= 0, and x= 3. Use your procedure simpleLimitTable as well as specific evaluations of h(x) to support your conclusions.

```
> simpleLimitTable(h,0,.01);
       -.100000          .010000
       -.010000          .000100
        .010000          .020000
        .100000          .200000
> simpleLimitTable(h,3,.0001);
       2.900000         5.800000
       2.990000         5.980000
       2.999000         5.998000
       2.999900         5.999800
       3.000100     10000.000000
       3.001000      1000.000000
       3.010000       100.000000
       3.100000        10.000000
> h(0); h(3);
```
0
6

Vertical asymptote at x= 0 and 3; Infinite Function as x approaches 3 from the right

Figure 94. Group N_4's analysis of mystery function h(x)

Problem #3 - Exploration of k(x)
Explain the behavior of the graph k(x) at x= 0, and x= 2. Use your procedure simpleLimitTable as well as specific evaluations of k(x) to support your conclusions.

```
> simpleLimitTable(k,0,.01);
       -.100000          .900000
       -.010000          .990000
        .010000         1.010000
        .100000         1.100000
> simpleLimitTable(k,2,.01);
       1.900000         2.900000
       1.990000         2.990000
       2.010000         3.010000
       2.100000         3.100000
```

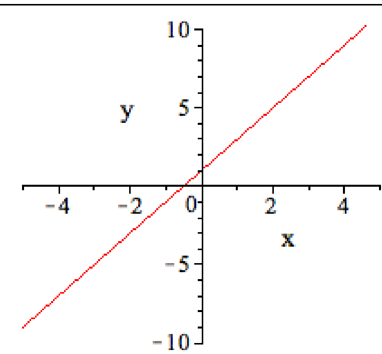
At zero the y-value is around 1 and at 2 the y-value is around 3, asymptotes??? VA at x=0 and 2

Figure 95. Group N_4's analysis of mystery function k(x)

Function m demonstrated a clearer understanding of limiting behavior. When asked to plot the function, the pair correctly produced a plot that appears linear (see Figure 96). Using the graph, they made a reasonable inference regarding the limit as x approaches 2. They subsequently utilized `simpleLimitTable` to the graph at x=2 to support their claim.

In Figure 97, the pair was then given the task of refining the closeness to x=2 by decreasing the step-size. Specifically, the students evaluated the function at points within 0.01, 0.001, 0.0001 of the given point. Unfortunately, the group did not use the tool as instructed at points within 0.0001 and so did not see and discrepant output. They mistakenly claim that this data supports the presence of a vertical asymptote, x=5. Evidently, they believe $\lim_{x\to 2} f(x) = L \;\; \Rightarrow \;\; y = L$ is a vertical asymptote; instead of $\lim_{x\to \pm\infty} f(x) = L \;\; \Rightarrow \;\; y = L$ is a vertical asymptote.

```
> plot ( m(x), x=-5..5, y=-10..10);
```



The limit at x=2 is 5.

The limit is 5.

```
> m(2);
                    3
> simpleLimitTable(m,2,.01);
          1.900000          4.800000
          1.990000          4.980000
          2.010000          5.020000
          2.100000          5.200000
```

Figure 96. Group N_4's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerance 0.01

<table>
<tr>
<td>

3. Use the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point.

<pre>
> simpleLimitTable(m,2,.001);
       1.900000   4.800000
       1.990000   4.980000
       1.999000   4.998000
       2.001000   5.002000
       2.010000   5.020000
       2.100000   5.200000
> simpleLimitTable(m,2,.01);
       1.900000   4.800000
       1.990000   4.980000
       2.010000   5.020000
       2.100000   5.200000
</pre>

4. Do you notice anything when you compare the results of step 2 and step 3?

Therefore x=5 must be a VA.

</td>
<td>

5. Using the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point.

<pre>
> simpleLimitTable(m,2,0.1);
       1.900000   4.800000
       2.100000   5.200000
> simpleLimitTable(m,2,0.001);
       1.900000   4.800000
       1.990000   4.980000
       1.999000   4.998000
       2.001000   5.002000
       2.010000   5.020000
       2.100000   5.200000
> simpleLimitTable(m,2,0.0001);
       1.900000   4.800000
       1.990000   4.980000
       1.999000   4.998000
       1.999900   4.000000
       2.000100   4.000000
       2.001000   5.002000
       2.010000   5.020000
       2.100000   5.200000
</pre>

6. Do you notice anything when you compare the results of steps 2, 3, and 4?

There is a VA at x=5, we think, but something weird is happening at x=1.9999.

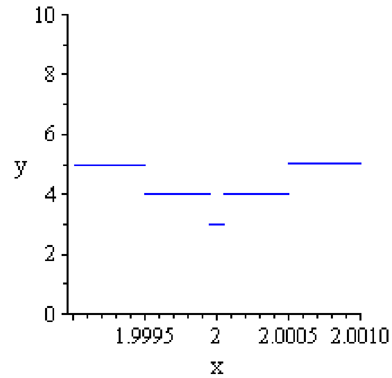
</td>
</tr>
</table>

Figure 97. Group N_4's response to Lab 2, Exploring mystery function m(x) near x=2 with tolerances 0.1, 0.01, 0.001, 0.0001 and 0.00001

Finally the group was asked to look as closely as necessary at the graph of m so as to explain the contradictory results. The pair produced the plot and explanation shown in Figure 98. The graph more clearly showed the unexpected behavior around x=2.

<table>
<tr><td>

```
>plot(m(x),x=1.999..2.001,y=0..10,
   color=blue,discont=true);
```

</td><td>
</td></tr>
<tr><td colspan="2">8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit? EXPLAIN HERE.

The limit tables made it seem like there was a Va at x=5, but there was really a piecewise function at x=2. You would have to go close enough so that you would get consistent answers from the limit table no matter how close to the x value you got.</td></tr>
</table>

Figure 98. Group N_4's response to Lab 2, Problem 8, exploring mystery function m(x) near x=2

From their response to Problem 8, the pair does understand the challenge-response nature of the limiting procedure. They also suggest that one would need to get arbitrarily close to x=2 to determine the true limiting value.

To summarize, the pair was able to use the `simpleLimitTable` tool and demonstrated an understanding of appropriate domain processes attaining understandings outlined in APOS steps 1, 2, 3a, and 3b. As a pair, the group continued to perform well utilizing the pair programming model again giving each other perfect peer evaluations.

Lab 2 Summary

Participants in class P, constructed and utilized the `simpleLimitTable` tool and participants in class N utilized the tool to explore the limiting behavior of functions whose graphical and algebraic representations were not available. This lab focused on developing an understanding of APOS steps 1, 2, 3a, and 3b for understanding the limit. Of specific interest are Steps 3a and 3b as they require the application of sequential argumentation.

Since students were hesitant to make such arguments in the first lab, tending to rely upon graphical and algebraic methods, the previously mentioned representational restrictions were deliberately imposed upon participants so as to limit their ability to use these types of arguments in favor of sequential arguments.

All the programming groups, class P, were able to successfully create the `simpleLimitTable` tool. They were also able to demonstrate its use. However, none of the programming groups successfully explored the mystery functions using the tool suggesting the programming groups saw the construction of the tool as the task rather than as a tool of exploration. Participants in class N, on the other hand, all attained the desired level of understanding within the APOS decomposition and actually utilized to tool to justify the behavior of the mystery functions.

The progression towards a complete understanding of limit within the APOS decomposition is shown in Figure 99. The vertical line highlights the intended level of attainment within the decomposition at the completion of the lab and the shading indicates the group's level of attainment.

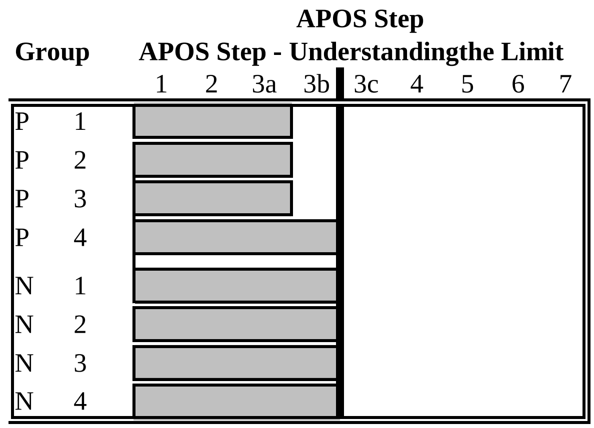


Figure 99. APOS Level of Attainment post Lab 2.
Shaded region represents actual attainment, vertical bar indicates the intended level of attainment.

It is with this lab that the two classes, P and N begin to diverge in their respective levels of attainment within the APOS framework.

Lab 3 Results

The third Maple lab took place during the tenth-twelfth weeks. Students were asked to create and/or utilize two tools, `leftLim` and `rightLim`, that produced one-sided limit tables similar to `simpleLimitTable` of lab two. The intent of these was to focus attention on individual left-hand and right-hand processes involved in ascertaining limiting behavior- a more careful exploration of the domain process. This understanding was subsequently used to foster an understanding of the coordination between the domain and resulting ranges processes as outlined in APOS Step 3c.

Once the tools were available, pairs were instructed to create hand-drawn sketches showing the limiting behavior of the mystery functions from lab two by inferring this behavior using the new tools coupled with function evaluation. Subsequently, the groups were given descriptions of the behavior of four new functions at and near specific points and asked to create functions having these stated behaviors. Students were to demonstrate and support the desired behaviors using the `leftLim` and `rightLim` tools.

This lab entailed several new concepts for both groups. The non-programming group developed an understanding of how to define piecewise functions in Maple and the programming group learned to create procedures with conditional behavior so as to implement piecewise function constructs. Additionally the programming group was introduced to the `for`-loop construct.

This lab was intended to focus upon the coordination of domain and range processes as expressed in steps 3a, 3b and 3c of the APOS decomposition and to address some notable issues observed in lab two. Students appeared to have some difficulty discussing one-sided behavior given two-sided information.

Following the lab, pairs completed a written post-lab activity that explored the group's ability to interpret limiting trends using tables of values. Pairs were shown eight `leftLim` and `rightLim` tables for unknown functions in which the domain and range sequences were manipulated in various ways. This written assignment asked students make conclusions regarding the functions' limiting behavior. In the second lab, students often failed to make specific inferences about limiting behavior. Thus, the intent of this lab was to expand upon the results from lab two; specifically, the degree to which they understood the coordination between the domain and range process of the limiting process, APOS step 3c.

Group P_1

The pair correctly constructed two procedures shown in Figure 100. Procedures `leftLim` and `rightLim` take three parameters, a function, *f*, a point, *a*, and a positive integer, *n*, indicating the degree of closeness to approach point *a*. In particular, if *n*=3, then the procedure would approach to within 10^{-3} of point a beginning at a distance of 0.1 and subsequently with decreasing powers of 10.

The pair developed an appropriate looping procedure which provided the correct

`leftLim` Procedure	**`rightLim` Procedure**

`> leftLim := proc( f, a, n)`
`    x := a-.1;`
`    for i from 1 to n do`
`      printf("%f %f\n", x, f(x));`
`      x := a-10^(-(i+1));`
`    end do:`
`end proc:`
	`> rightLim := proc( f, a, n)`
	`    x := a+.1;`
	`    for i from 1 to n do`
	`      printf("%f %f\n", x, f(x));`
	`      x := a+10^(-(i+1));`
	`    end do:`
	`end proc:`

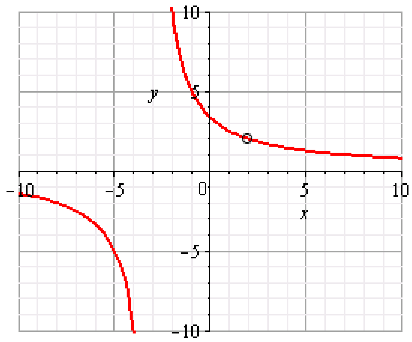
Figure 100. Group P_1's Definition of `leftLim` and `rightLim` procedures in Lab 3

sequence of function evaluations. The specific loop suggests an interiorization of the domain process utilized in the limiting process, APOS step 2 and 3a.

Once these tools were developed, the pair utilized them to infer the behavior of the four mystery functions studied in the previous lab. They explored the behavior of these functions at specified points and produced hand-drawn graphs of the behavior they inferred. These inferences and supporting tool use are shown below. As in the second lab, the students did not have access to actual plots of the mystery functions. These graphs are shown in Figure 101, however, for comparative purposes.

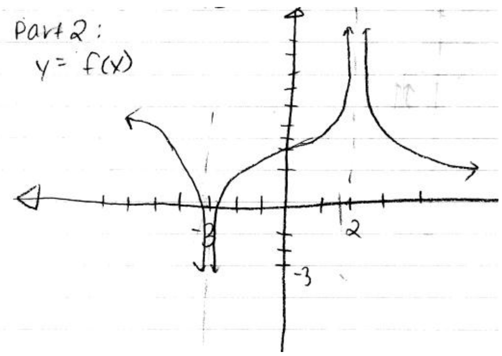
In this step, the pair unnecessarily created a new function, `f(x)`, and used that function at point x=-3 (left) whereas at x=2 (right) the actual mystery function was utilized. To Maple, "`f(x):=`" and "`f:=`" are different function definitions. As can be seen in Figure 101 the function `f(x)` they created was the squaring function. Thus the pair is using two different functions in their analysis, the function of their own devising f(x) is utilized at x=-3 and the intended mystery function, f., is used at x=2. Neither of the pair noticed anything surprising about these results. More troubling is the resulting sequences of evaluations produced by the tools are in no way suggestive of the behavior shown in their graph. Evaluations near x=-3 do not suggest asymptotic behavior, nor do the evaluations near x=2. It is unclear where the graph they produced originated.

Actual Mystery Function $f(x)$



Hole at x=2, Vertical asymptote at x= -3.

Inferred Behavior



Support at x=-3 using `leftLim` and `rightLim`

```
> f(x) := proc( x )
   if type(x, realcons) then
      x^2;
   else
      'fn(x)';
   end if:
end proc:

> leftLim( f(x), -3, 6):
-3.100000 9.610000
-3.010000 9.060100
-3.001000 9.006001
-3.000100 9.000600
-3.000010 9.000060
-3.000001 9.000006
> rightLim( f(x), -3, 6):
-2.900000 8.410000
-2.990000 8.940100
-2.999000 8.994001
-2.999900 8.999400
-2.999990 8.999940
-2.999999 8.999994
```

Support at x=2 using `leftLim` and `rightLim`

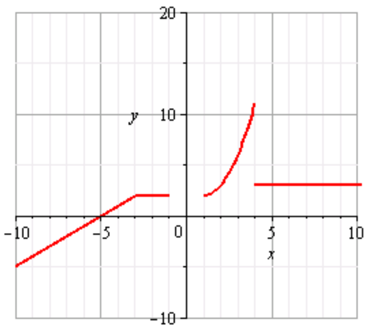
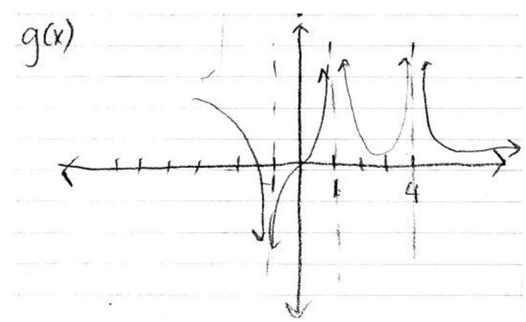
```
> leftLim ( f, 2, 6):
1.900000 2.040816
1.990000 2.004008
1.999000 2.000400
1.999900 2.000040
1.999990 2.000004
1.999999 2.000000
> rightLim ( f, 2, 6):
2.100000 1.960784
2.010000 1.996008
2.001000 1.999600
2.000100 1.999960
2.000010 1.999996
2.000001 2.000000
```

Figure 101. Group P_1's Analysis of mystery function f(x)

These issues highlight two major obstacles challenging the programming groups, one being the challenge of developing basic programming skill and the other learning to effectively utilize the tools once developed. While the group is able to properly implement the tool, their lack of understanding of the naming conventions of Maple caused them to analyze a function different from the one they intended. However, even

with an effective tool at their disposal, the group is unable to offer an appropriate analysis of the functions behavior from the data they did collect; they are unable to use the tool.

The pair had similar issues with the function g graphs as shown in Figure 102. This time the graph did not make (accidental) use of the actual mystery function g. The resulting output, shown in Figure 102, is not suggestive of the behavior neither of their hand-drawn graph nor of the graph of g (x).

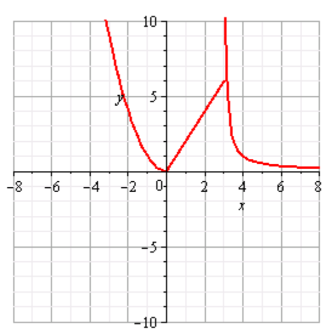
Actual Mystery Function $g(x)$	**Inferred Behavior**
 Jump at x=4, no left limit at x=1, no right limit at x=-1.	

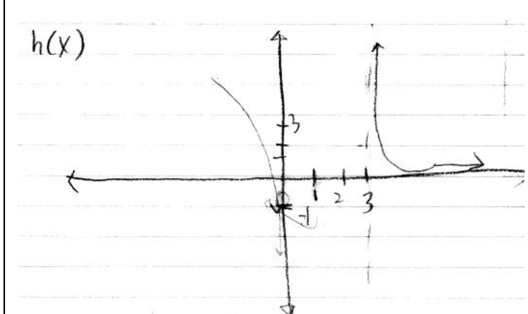
Support at x=-1 using `leftLim` and `rightLim`
 	
`> g(x) := proc( x )`
`   if type(x, realcons) then`
`      x^2;`
`   else`
`      'fn(x)';`
`   end if:`
`end proc:`
 	
`> leftLim ( g(x), -1, 5):`
`-1.100000 1.210000`
`-1.010000 1.020100`
`-1.001000 1.002001`
`-1.000100 1.000200`
`-1.000010 1.000020`
`> rightLim ( g(x), -1, 5):`
`-0.900000 0.810000`
`-0.990000 0.980100`
`-0.999000 0.998001`
`-0.999900 0.999800`
`-0.999990 0.999980`
	Support at x=1, 4 using `leftLim` and `rightLim`
	`> leftLim ( g(x), 1, 5):`
	`0.900000 0.810000`
	`0.990000 0.980100`
	`0.999000 0.998001`
	`0.999900 0.999800`
	`0.999990 0.999980`
	`> rightLim ( g(x), 1, 5):`
	`1.100000 1.210000`
	`1.010000 1.020100`
	`1.001000 1.002001`
	`1.000100 1.000200`
	`1.000010 1.000020`
	`> leftLim (g(x), 4, 5):`
	`3.900000 15.210000`
	`3.990000 15.920100`
	`3.999000 15.992001`
	`3.999900 15.999200`
	`3.999990 15.999920`
	`> rightLim (g(x), 4, 5):`
	`4.100000 16.810000`
	`4.010000 16.080100`
	`4.001000 16.008001`
	`4.000100 16.000800`
	`4.000010 16.000080`

Figure 102. Group P_1's analysis of mystery function g(x)

For functions h and k, shown in Figure 103 and Figure 104, the pair continued to create a new squaring function and utilize that function. For function h, the pair only explored the behavior on one-side of the specified points, x=0 and x=3. Again the data does not support their resulting graph.

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple that have specified behavior and utilize the `leftLim` and `rightLim` procedures to support their claims. The first function they were asked to create, b(x), is shown in Figure 105. As shown, the pair did not accomplish either the task of creating a hole at x=2 or creating a vertical asymptote at x=-1. Neither the graph nor the limit tables output support the claimed behavior.

<table>
<tr><th>Actual Mystery Function $h(x)$</th><th>Inferred Behavior</th></tr>
<tr><td>

Corner at x=0 and a Vertical asymptote at x=3.</td><td>
</td></tr>
<tr><td>Support at x=0 using only <code>leftLim</code>

<code>> leftLim (h(x), 0, 5):
-0.100000 0.010000
-0.010000 0.000100
-0.001000 0.000001
-0.000100 0.000000
-0.000010 0.000000</code></td><td>Support at x=3 using only <code>rightLim</code>

<code>> rightLim (h(x), 3, 5):
3.100000 9.610000
3.010000 9.060100
3.001000 9.006001
3.000100 9.000600
3.000010 9.000060</code></td></tr>
</table>

Figure 103. Group P_1's analysis of mystery function h(x)

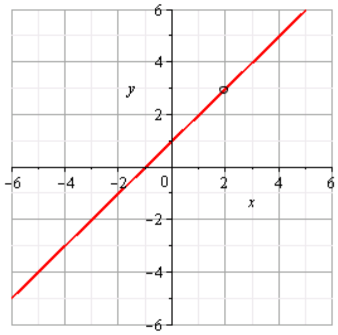
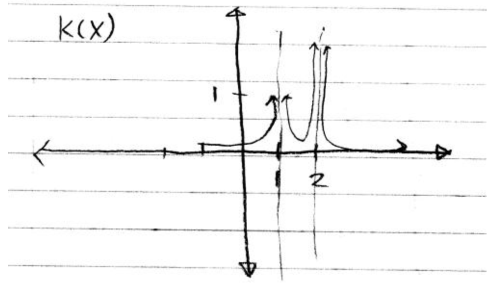
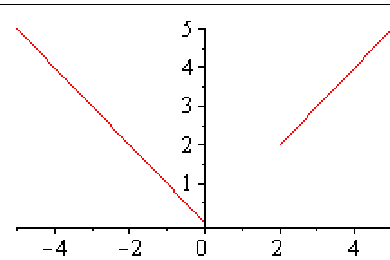
Actual Mystery Function $k(x)$	Inferred Behavior
 Hole at x=2.	
Support at x=1 using leftLim and rightLim > leftLim (k(x), 1, 5): 0.900000 0.810000 0.990000 0.980100 0.999000 0.998001 0.999900 0.999800 0.999990 0.999980 > rightLim (k(x), 1, 5): 1.100000 1.210000 1.010000 1.020100 1.001000 1.002001 1.000100 1.000200 1.000010 1.000020	Support at x=2 using leftLim and rightLim > leftLim (k(x), 2, 5): 1.900000 3.610000 1.990000 3.960100 1.999000 3.996001 1.999900 3.999600 1.999990 3.999960 > rightLim (k(x), 2, 5): 2.100000 4.410000 2.010000 4.040100 2.001000 4.004001 2.000100 4.000400 2.000010 4.000040

Figure 104. Group P_1's analysis of mystery function k(x)

Desired Behavior	Function definition	Plot

Hole at x= 2
Vertical Asymptote at x=-1
	b := proc(x)
	if type(x, realcons) then
	if x>=2 then
	x;
	elif x<0 then
	-x;
	else
	undefined;
	end if:
	else
	'b(x)';
	end if:
	end proc:
		

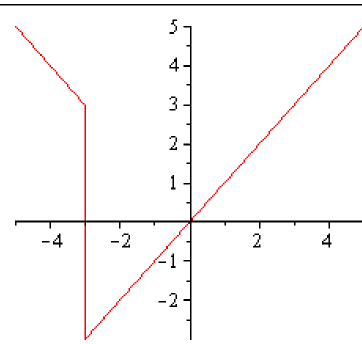
Supporting data produced with tools.	> leftLim (b, 2, 4): 1.900000 NaN 1.990000 NaN 1.999000 NaN 1.999900 NaN > rightLim (b, 2, 4): 2.100000 2.100000 2.010000 2.010000 2.001000 2.001000 2.000100 2.000100	> leftLim (b, -1, 4): -1.100000 1.100000 -1.010000 1.010000 -1.001000 1.001000 -1.000100 1.000100 > rightLim (b, -1, 4): -0.900000 0.900000 -0.990000 0.990000 -0.999000 0.999000 -0.999900 0.999900

Figure 105. Group P_1's constructed function b(x) for Lab 3

For function c (see Figure 106), the pair again produced none of the requested behavior. From a programming perspective, the pair fails to recognize that the final `else` clause is unnecessary as the prior `if` statements address all possible values for x.

For function d (see Figure 107), the pair was able to accomplish the requested domain restriction by a suitable conditional statement as well as produce and support appropriate limiting behavior.

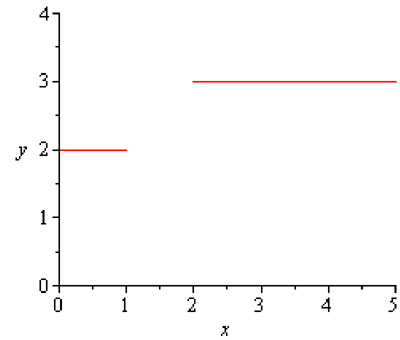
Desired Behavior	**Function definition**	**Plot**

Jump discontinuity at x= -1
$\lim_{x\to -3^-} c(x) = +\infty$
$\lim_{x\to -3^+} c(x) = +\infty$
	`c := proc(x)`
	`  if type(x, realcons) then`
	`      if x>-3 then`
	`         x;`
	`     elif x<-3 then`
	`         -x;`
	`     else`
	`         undefined;`
	`      end if:`
	`   else`
	`       'c(x)';`
	`   end if:`
	`end proc:`
		

Supporting data produced with tools.	`> rightLim (c,-1,4):`
	`-0.900000 -0.900000`
	`-0.990000 -0.990000`
	`-0.999000 -0.999000`
	`-0.999900 -0.999900`	

Figure 106. Group P_1's constructed function c(x) in Lab 3

Desired Behavior	**Function definition**	**Plot**

Undefined on [1,2]
$\lim\limits_{x\to 1^-} d(x)=2$
$\lim\limits_{x\to 2^+} d(x)=3$
	`d := proc(x)`
	`  if type(x, realcons)`
	`then`
	`      if x>2 then`
	`        3;`
	`     elif x<1 then`
	`        2;`
	`     else`
	`        undefined;`
	`      end if:`
	`   else`
	`       'd(x)';`
	`   end if:`
	`end proc:`
		

Supporting data produced with tools.	`> leftLim (d,1,4):`
	`0.900000 2.000000`
	`0.990000 2.000000`
	`0.999000 2.000000`
	`0.999900 2.000000`
	`> rightLim (d,1,4):`
	`1.100000 NaN`
	`1.010000 NaN`
	`1.001000 NaN`
	`1.000100 NaN`
		`> leftLim (d,2,4):`
		`1.900000 NaN`
		`1.990000 NaN`
		`1.999000 NaN`
		`1.999900 NaN`
		`> rightLim (d,2,4):`
		`2.100000 3.000000`
		`2.010000 3.000000`
		`2.001000 3.000000`
		`2.000100 3.000000`

Figure 107. Group P_1's constructed function d(x) in Lab 3

Function e (see Figure 108) also failed to meet any of the criteria requested. The pair did not understand the way conditional statements are evaluated as with function c. In particular, any input larger than -2 will return the value using the function, e(x)=x. The remaining conditionals will never be evaluated. It is also not clear why the students thought that infinite limiting behavior occurs at x=-2. They appear to misunderstand this type of limiting behavior as in function b(x) earlier.

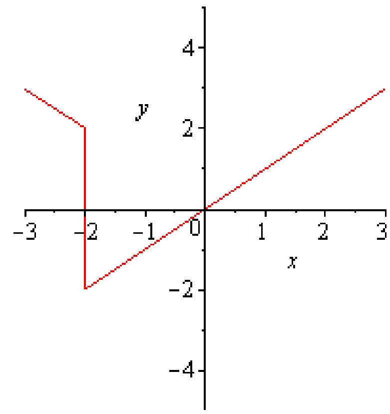
<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr>
<td>$\lim_{x \to 1} e(x) = 3$
$e(1) = 5$
$\lim_{x \to -2^-} e(x) = +\infty$
$\lim_{x \to -2^+} e(x) = -\infty$</td>
<td><pre>
e := proc(x)
  if type(x, realcons) then
      if x>-2 then
         x;
      elif x=1 then
         5;
      elif x>1 then
         3;
      elif x<-2 then
         -x;
      else
         undefined;
      end if:
   else
       'e(x)';
   end if:
end proc:
</pre></td>
<td>
</td>
</tr>
</table>

Figure 108. Group P_1's constructed function e(x) in Lab 3

Following the lab, the group completed a written post-lab activity to explore the degree to which they understood the coordination between the domain and range process of the limiting process. The responses given further confirm earlier observations. For example, in situations where the domain process and range processes are coordinated, the pair makes appropriate inferences about the limiting behavior. However, when the domain process is randomized, the pair consistently bases their conclusions on the range behavior alone (see Figure 109).

Table 2		**Table 3**		**Table 6**	
x	f(x)	x	f(x)	x	f(x)
2.10000000	4.87930340	2.10000000	4.87930340	2.10000000	7.94010000
2.01000000	4.08722195	2.01000000	4.08722195	2.01000000	5.49230400
2.00100000	4.00871339	2.00100000	4.00871339	2.00100000	5.88537600
2.00010000	4.00087125	2.00010000	4.00087125	2.00010000	7.47392100
2.00001000	4.00008712	2.00001000	4.00008712	2.00001000	7.74976400
2.00000100	4.00000871	2.00000100	4.00000871	2.00000100	5.79644900
2.00000010	4.00000087	2.00000010	4.00000087	2.00000010	5.32522500
2.00000001	4.00000009	2.00000001	4.00000009	2.00000001	3.48168900
x	f(x)	x	f(x)	x	f(x)
1.68000000	6.85900000	1.42100000	5.41468922	1.90000000	-2.72000000
1.87800000	7.88059900	1.17400000	4.19665340	1.99000000	-2.16200000
1.62000000	7.98800600	1.80400000	1.29502900	1.99900000	-1.09400000
1.16200000	7.99880006	1.15200000	2.86328800	1.99990000	-2.13800000
1.40500000	7.99988000	1.79700000	1.65979750	1.99999000	-2.28600000
1.71600000	7.99998800	1.56900000	2.20207390	1.99999900	-2.51000000
1.70100000	7.99999880	1.64100000	2.06293342	1.99999990	-2.68600000
1.29000000	7.99999988	1.11000000	5.73533900	1.99999999	-2.77000000
Conclusions: f(2)=4		**Conclusions:**		**Conclusions:**	
$\lim_{x\to 2^-} f(x)=8$		$\lim_{x\to 2^-} f(x)=\infty$		$\lim_{x\to 2^-} f(x)=DNE$	
$\lim_{x\to 2^+} f(x)=4$		$\lim_{x\to 2^+} f(x)=4$		$\lim_{x\to 2^+} f(x)=DNE$	

Figure 109. Group P_1's response to Post-lab 3 questions

In tables 2 and 3, the fact that the domain process does not approach any particular value from one side missed and incorrect inferences are made, i.e. $\lim_{x\to 2^-} f(x)=8$ and $\lim_{x\to 2^-} f(x)=\infty$. Yet, when the domain process fails to converge but the range process does, an appropriate conclusion is often made. This suggests that the group is focusing on the range process alone. Additionally, with table three the group makes the strange comment that $\lim_{x\to 2^-} f(x)=\infty$ when there does not appear to be any asymptotic behavior present.

This pair has misunderstandings relating to infinite limits as well as several programming related issues. The pair demonstrated an understanding of the domain process, APOS 3a, but clearly had issues related to the range process, APOS 3b. Therefore, an understanding of the coordinated relationship between these two processes in not in evidence for this group, APOS 3c.

One group member began having a large number of absences beginning with this lab. This put additional responsibility on their partner, who continued to work on the lab in her absence. Ultimately, the missed work was completed but apparently such group instabilities profoundly affected the group's continuity and understanding. In the peer reviews, both students gave each other perfect review- both expressing the helpfulness of the other partner.

Group P_2

The pair correctly constructed two procedures, `leftLim` and `rightLim`, shown in Figure 110, that take three parameters, a function, *f*, a point, *a*, and a positive integer, *n*, indicating the degree of closeness to approach point *a*. Noticeably, the group made use of local variables, *x* and *i*, suggesting understanding of variables and their scope.

`leftLim` Procedure

```
> leftLim := proc( f, a, n)
 local x,i;
 x:= a-0.1;
 for i from 1 to n do
  printf("%15.8f
%15.8f\n",x,f(x));
  x:= a-10^(-(i+1));
 end do:
end proc:
```

`rightLim` Procedure

```
> rightLim := proc( f, a, n)
 local x,i;
 x := a+0.1;
 for i from 1 to n do
  printf("%15.8f%15.8f\n",x,f(x));
  x:= a+10^(-(i+1));
 end do:
end proc:
```

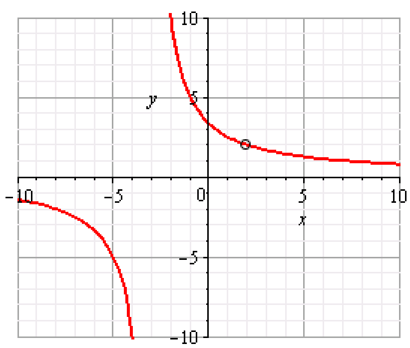
Figure 110. Group P_2's Definition of `leftLim` and `rightLim` procedures in Lab 3

The pair also developed an appropriate looping procedure which produced the correct sequence of function evaluations. The specific loop suggests an interiorization of the domain process utilized in the limiting process, APOS step 2 and 3a.

Once these tools were developed, the pair utilized them to infer the behavior of the four mystery functions from the previous lab. As before, the students did not have access to actual graphs of the mystery functions. These graphs are shown in Figure 111 for comparative purposes.

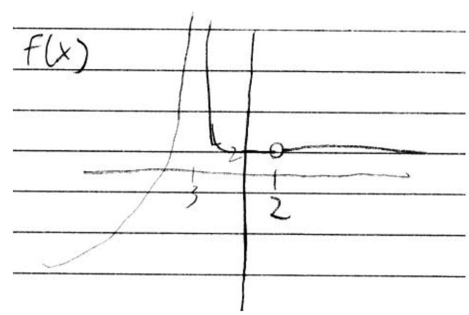
In this step, the pair correctly inferred the behavior on from the right as one approaches x=-3. The behavior on the left side of x=-3 was not correctly inferred, however. While the graph does have asymptotic behavior on both sides of x=-3, it

Actual Mystery Function $f(x)$



Hole at x=2, Vertical asymptote at x= -3.

Inferred Behavior



Support at x=-3 using `leftLim` and `rightLim`

```
> leftLim(f, -3, 5):
    -3.10000000    -100.00000000
    -3.01000000   -1000.00000000
    -3.00100000  -10000.00000000
    -3.00010000 -100000.00000000
    -3.00001000 -1000000.00000000
> rightLim(f,-3,5):
    -2.90000000     100.00000000
    -2.99000000    1000.00000000
    -2.99900000   10000.00000000
    -2.99990000  100000.00000000
    -2.99999000 1000000.00000000
```

Support at x=2 using `leftLim` and `rightLim`

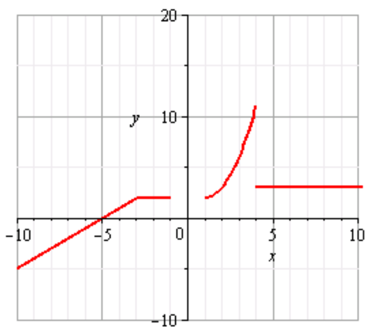
```
> leftLim(f,2,5):
     1.90000000        2.04081633
     1.99000000        2.00400802
     1.99900000        2.00040008
     1.99990000        2.00004000
     1.99999000        2.00000400
> rightLim(f,2,5):
     2.10000000        1.96078431
     2.01000000        1.99600798
     2.00100000        1.99960008
     2.00010000        1.99996000
     2.00001000        1.99999600
```

Figure 111. Group P_2's analysis of mystery function f(x)

decreases to $-\infty$ on the left rather than $+\infty$ as shown in their graph. The pair also correctly inferred the presence of a hole at x=2 but did not perform evaluation at x=2 to see whether the function possessed a value at that point. Thus there inference is not completely justified.

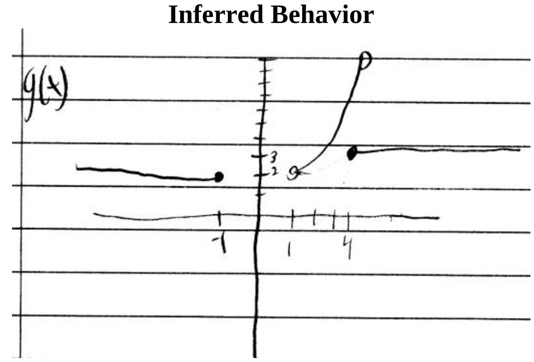
The pair had similar success with the remaining graphs. Their graph of g, shown in Figure 112, is nearly complete, except that they never performed evaluations at x=-1,1,

Actual Mystery Function $g(x)$



Jump at x=4, no left limit at x=1, no right limit at x=-1.

Inferred Behavior



Support at x=-1, 1 using `leftLim` and `rightLim`

```
> leftLim(g,-1,4):
    -1.10000000      2.00000000
    -1.01000000      2.00000000
    -1.00100000      2.00000000
    -1.00010000      2.00000000
> rightLim(g,-1,4):
    -0.90000000             NaN
    -0.99000000             NaN
    -0.99900000             NaN
    -0.99990000             NaN
> leftLim(g,1,4):
     0.90000000             NaN
     0.99000000             NaN
     0.99900000             NaN
     0.99990000             NaN
> rightLim(g,1,4):
     1.10000000      2.01000000
     1.01000000      2.00010000
     1.00100000      2.00000100
     1.00010000      2.00000001
```

Support at x=4 using `leftLim` and `rightLim`

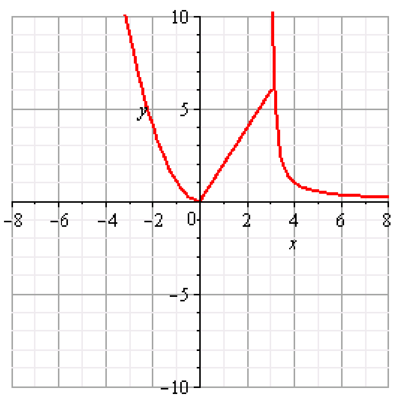
```
> leftLim(g,4,4):
     3.90000000      10.41000000
     3.99000000      10.94010000
     3.99900000      10.99400100
     3.99990000      10.99940001
> rightLim(g,4,4):
     4.10000000       3.00000000
     4.01000000       3.00000000
     4.00100000       3.00000000
     4.00010000       3.00000000
```

Figure 112. Group P_2's analysis of mystery function g(x) in Lab 3

and 4 to determine where, and if, such points should be included. They made appropriate inferences using the data provided by the tools.

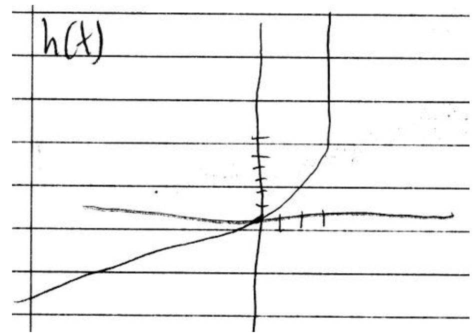
Function h is shown in Figure 113. While the pair makes appropriate inferences, there were some oversights. On the left side of x=0, they pair incorrectly inferred increasing behavior rather than decreasing and at x=3, they incorrectly discern behavior on the right.

Actual Mystery Function $h(x)$



Corner at x=0 and a Vertical asymptote at x=3.

Inferred Behavior



Support at x=0 using `leftLim` and `rightLim`

```
> leftLim(h,0,6):
    -0.10000000       0.01000000
    -0.01000000       0.00010000
    -0.00100000       0.00000100
    -0.00010000       0.00000001
    -0.00001000       0.00000000
    -0.00000100       0.00000000
> rightLim(h,0,6):
     0.10000000       0.20000000
     0.01000000       0.02000000
     0.00100000       0.00200000
     0.00010000       0.00020000
     0.00001000       0.00002000
     0.00000100       0.00000200
```

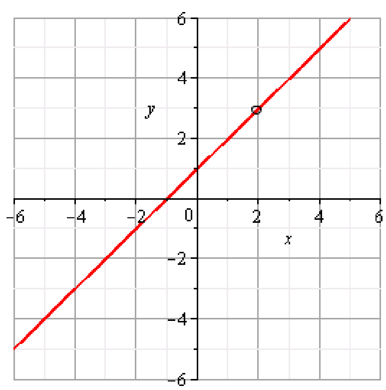
Support at x=3 using `leftLim` and `rightLim`

```
> leftLim(h,3,6):
  2.90000000        5.80000000
  2.99000000        5.98000000
  2.99900000        5.99800000
  2.99990000        5.99980000
  2.99999000        5.99998000
  2.99999900        5.99999800
> rightLim(h,3,6):
  3.10000000       10.00000000
  3.01000000      100.00000000
  3.00100000     1000.00000000
  3.00010000    10000.00000000
  3.00001000   100000.00000000
  3.00000100  1000000.00000000
```

Figure 113. Group P_2's analysis of mystery function h(x) in Lab 3

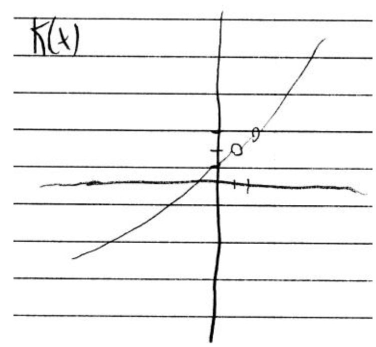
On function k, their graph is nearly correct except that there is no hole at x=1 (see Figure 114). This is a recurrent issue; they had the same issue previously with functions f, and g. Apparently, they don't see the necessity of evaluating the function at the point in determining whether a hole is present. Their use of the tools, however, and the resulting inferences are largely accurate and relevant.

Actual Mystery Function $k(x)$



Hole at x=2.

Inferred Behavior



Support at x=1 using `leftLim` and `rightLim`

```
> leftLim(k,1,6):
      0.90000000        1.90000000
      0.99000000        1.99000000
      0.99900000        1.99900000
      0.99990000        1.99990000
      0.99999000        1.99999000
      0.99999900        1.99999900
> rightLim(k,1,6):
      1.10000000        2.10000000
      1.01000000        2.01000000
      1.00100000        2.00100000
      1.00010000        2.00010000
      1.00001000        2.00001000
      1.00000100        2.00000100
```

Support at x=2 using `leftLim` and `rightLim`

```
> leftLim(k,2,6):
      1.90000000        2.90000000
      1.99000000        2.99000000
      1.99900000        2.99900000
      1.99990000        2.99990000
      1.99999000        2.99999000
      1.99999900        2.99999900
> rightLim(k,2,6):
      2.10000000        3.10000000
      2.01000000        3.01000000
      2.00100000        3.00100000
      2.00010000        3.00010000
      2.00001000        3.00001000
      2.00000100        3.00000100
```

Figure 114. Group P_2's analysis of mystery function k(x) in Lab 3

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple that have specified behavior. As shown in Figure 115, the pair accomplished both the task of creating a hole at x=2 and creating a vertical asymptote at x=-1. The resulting limit tables support the hole at x=2 but not the asymptotic behavior at x=-1. Additionally, they did ensure that the function failed to have a value at x=2 which had been an oversight in prior analyses.

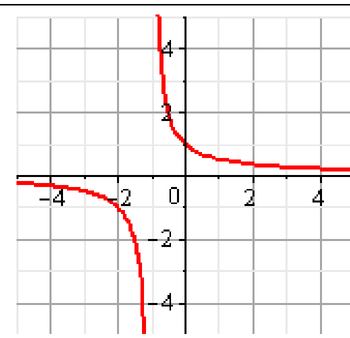
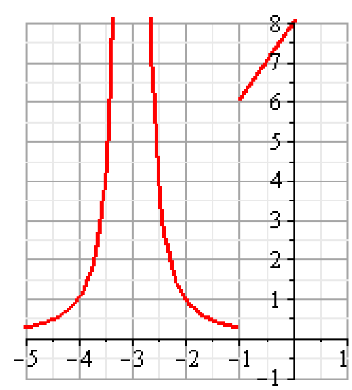
<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr><td>Hole at x= 2
Vertical Asymptote at x=-1</td><td><code>> b := proc(x)
   if type(x, realcons) then
      (x-2)/(x^2-x-2);
   else
      'b(x)';
   end if:
end proc:</code></td><td>
</td></tr>
<tr><td>Supporting data produced with tools.</td><td colspan="2"><code>> b(2);
Error, (in b) numeric exception: division by zero
> leftLim(b, 2, 5):
1.90000000   0.34482759
1.99000000   0.33444816
1.99900000   0.33344448
1.99990000   0.33334444
1.99999000   0.33333444
> rightLim(b,2,5):
2.10000000   0.32258065
2.01000000   0.33222591
2.00100000   0.33322226
2.00010000   0.33332222
2.00001000   0.33333222</code></td></tr>
</table>

Figure 115. Group P_2's constructed function b(x) in Lab 3

Again, for function c, in Figure 116, the pair produced of procedure with all requested behavior, produced a graph, and made accurate inferences using the tools. Function d, shown in Figure 117, possessed the requested domain restriction but did not have the requested limiting behavior. It appears the group may have copied function c with the intent of modifying it to have the desired behavior but this modification was never completed. Function e, shown in Figure 118, nearly satisfied all the criteria specified and was accurately supported with the application of `leftLim` and `rightLim` and evaluation. The only oversight was that the asymptotic behavior around x=-2 is reversed. Specifically, the pair has $\lim_{x \to -2^-} e(x) = -\infty$ and $\lim_{x \to -2^+} e(x) = +\infty$.

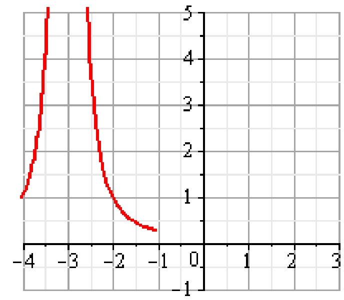
Desired Behavior	**Function definition**	**Plot**

Jump discontinuity at x= -1
$\lim_{x \to -3^-} c(x) = +\infty$
$\lim_{x \to -3^+} c(x) = +\infty$
	`> c := proc(x)`
	`if type(x, realcons) then`
	`if x>=-1 then`
	`2*x+8;`
	`elif x<-1 then`
	`1/(x+3)^2;`
	`else`
	`undefined;`
	`end if:`
	`else`
	`'c(x)';`
	`end if:`
	`end proc:`
		

Supporting data produced with tools.	`> leftLim(c, -3, 5):`
	`-3.100000 100.000000`
	`-3.010000 10000.000000`
	`-3.001000 1000000.000000`
	`-3.000100 100000000.000000`
	`-3.000010 10000000000.000000`
	
	`> rightLim(c,-3,5):`
	`-2.900000 100.000000`
	`-2.990000 10000.000000`
	`-2.999000 1000000.000000`
	`-2.999900 100000000.000000`
	`-2.999990 10000000000.000000`
		`> leftLim(c, -1, 5):`
		`-1.100000 0.27700831`
		`-1.010000 0.25251888`
		`-1.001000 0.25025019`
		`-1.000100 0.25002500`
		`-1.000010 0.25000250`
		`> rightLim(c,-1,5):`
		`-0.900000 6.200000`
		`-0.990000 6.020000`
		`-0.999000 6.002000`
		`-0.999900 6.000200`
		`-0.999990 6.000020`

*Figure 116.*Group P_2's construction of function c(x) in Lab 3

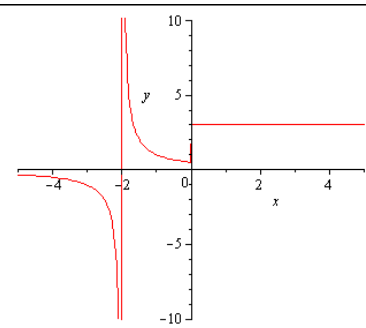
Desired Behavior	**Function definition**	**Plot**

Undefined on [1,2]
$\lim_{x\to1^-} d(x) = 2$
$\lim_{x\to2^+} d(x) = 3$
	`> d := proc(x)`
	`if type(x, realcons) then`
	`if x>=-1 then`
	`;`
	`elif x<-1 then`
	`1/(x+3)^2;`
	`else`
	`undefined;`
	`end if:`
	`else`
	`'d(x)';`
	`end if:`
	`end proc:`
		

Supporting data produced with tools.	None	

Figure 117. Group P_2's construction of function d(x) in Lab 3

Desired Behavior	**Function definition**	**Plot**

$\lim_{x\to1} e(x) = 3$
$e(1) = 5$
$\lim_{x\to-2^-} e(x) = +\infty$
$\lim_{x\to-2^+} e(x) = -\infty$
	`> e := proc(x)`
	`if type(x, realcons) then`
	`if x=1 then`
	`5;`
	`elif x>=0 then`
	`3;`
	`elif x <0 then`
	`(1)/(x+2);`
	`else`
	`undefined;`
	`end if:`
	`else`
	`'e(x)';`
	`end if:`
	`end proc:`
		

Supporting data produced with tools.	`> e(1);`
	`5`
	`> leftLim(e, 1, 5):`
	`0.900000 3.000000`
	`0.990000 3.000000`
	`0.999000 3.000000`
	`0.999900 3.000000`
	`0.9999900 3.000000`
	`> rightLim(e,1,5):`
	`1.100000 3.000000`
	`1.010000 3.000000`
	`1.001000 3.000000`
	`1.000100 3.000000`
	`1.000010 3.000000`
		`> leftLim(e, -2, 5):`
		`-2.100000 -10.000000`
		`-2.010000 -100.000000`
		`-2.001000 -1000.000000`
		`-2.000100 -10000.000000`
		`-2.000010 -100000.000000`
		`> rightLim(e,-2,5):`
		`-1.900000 10.000000`
		`-1.990000 100.000000`
		`-1.999000 1000.000000`
		`-1.999900 10000.000000`
		`-1.999990 100000.000000`

Figure 118.Group P_2's construction of function e(x) in Lab 3

Following the lab, the group completed a written post-lab activity to explore the degree to which they understood the coordination between the domain and range process of the limiting process. This group failed to submit this work.

This pair seemed to possess a very good understanding of the limit process and was able to construct and utilize the tools effectively. Based upon there inferences within the lab itself, the pair demonstrated an understanding of the domain process, APOS 3a, as well as the need for a corresponding range process, APOS 3b. An understanding of the coordinated relationship between these two processes is suggested by their consistently accurate inferences but this is not confirmed by post-lab responses as none were submitted. Nevertheless, there is support for this group's attainment of APOS Step 3c. The group gave each other perfect peer reviews and commented that they continue to enjoy working together and that the labs are fun.

Group P_3

The pair correctly constructed two procedures, `leftLim` and `rightLim` shown in Figure 119. As can been seen in the procedure declarations, this group made use of local variables, x and i, suggesting a deeper understanding of variables and their scope.

`leftLim` Procedure	**`rightLim` Procedure**

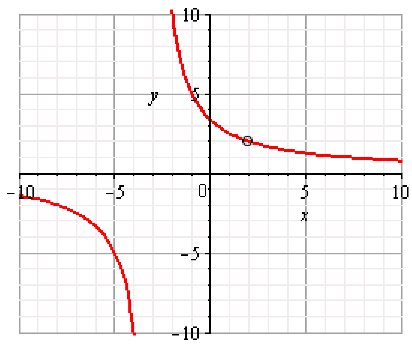
`> leftLim := proc( f, a, n)`
`local x, i;`
` x:= a-0.1;`
` for i from 1 to n do`
`  printf("%15.8f`
`%15.8f\n",x,f(x));`
`  x:=a-10^(-(i+1));`
` end do:`
`end proc:`
	`> rightLim := proc( f, a, n)`
	` local x, i;`
	` x:= a+0.1;`
	` for i from 1 to n do`
	`  printf("%15.8f %15.8f\n",x,f(x));`
	`  x:= a+10^(-(i+1));`
	` end do:`
	`end proc:`

Figure 119. Group P_3's implementation of `leftLim` and `rightLim` procedures

Additionally, the pair developed an appropriate looping procedure which provided the correct sequence of function evaluations. The specific loop suggests an interiorization of the domain process utilized in the limiting process, APOS step 2 and 3a.

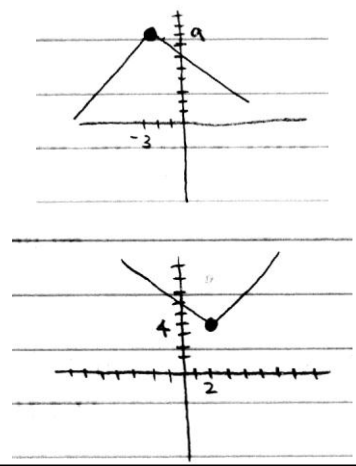
Once these tools were developed, the pair attempted to utilize them to infer the behavior of the four mystery functions from Lab 2. As shown in Figure 120, the pair created a squaring function named f(x) and proceeded to analyze this new function rather

Actual Mystery Function $f(x)$



Hole at x=2, Vertical asymptote at x= -3.

Inferred Behavior



Support at x=-3 using `leftLim` and `rightLim`

```
> f(x) := proc( x )
   if type(x, realcons) then
      x^2;
   else
      'f(x)';
   end if:
end proc:

> leftLim (f(x),-3,4):
   -3.10000000      9.61000000
   -3.01000000      9.06010000
   -3.00100000      9.00600100
   -3.00010000      9.00060001
> rightLim (f(x),-3,4):
   -2.90000000      8.41000000
   -2.99000000      8.94010000
   -2.99900000      8.99400100
   -2.99990000      8.99940001
```

Support at x=2 using `leftLim` and `rightLim`

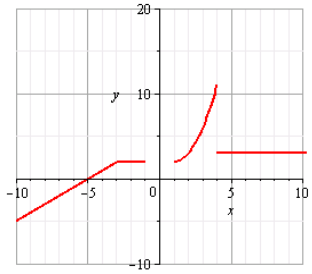
```
> leftLim (f(x),2,4):
     1.90000000       3.61000000
     1.99000000       3.96010000
     1.99900000       3.99600100
     1.99990000       3.99960001
> rightLim (f(x),2,4):
     2.10000000       4.41000000
     2.01000000       4.04010000
     2.00100000       4.00400100
     2.00010000       4.00040001
```

Figure 120. Group P_3's analysis of mystery function f(x) in Lab 3

than the intended mystery function, f. This group did, however, consistently use the new function but was unable to interpret the left-hand limit data. They erroneously inferred the function increases from the left of x=-3 and deceases from the left of x=2 when in fact the function increases; they demonstrate a clear lack of coordination between the domain and range processes.

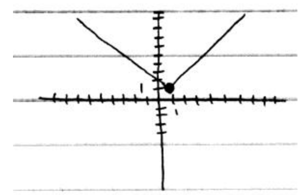
The pair had similar success with mystery function g as shown in Figure 121. As before, using the incorrect function, the pair inferred the wrong right behavior at x=-1, the wrong left behavior at x=1, and the wrong right behavior at x=4.

Actual Mystery Function $g(x)$

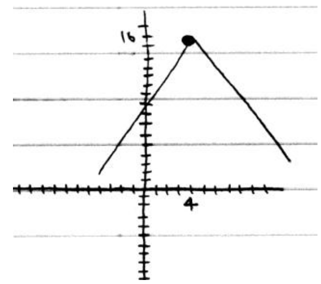


Jump at x=4, no left limit at x=1, no right limit at x=-1.

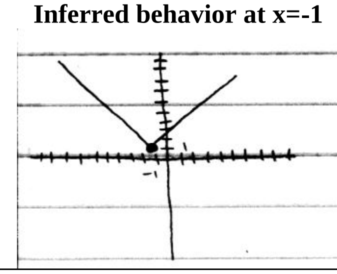
Inferred Behavior at x=1



Inferred Behavior at x=4



Inferred behavior at x=-1



Support at x=-1, 1 using `leftLim` and `rightLim`

```
> g(x) := proc( x )
   if type(x, realcons) then
      x^2;
   else
      'g(x)';
   end if:
end proc:
> leftLim (g(x),-1,4):
    -1.10000000      1.21000000
    -1.01000000      1.02010000
    -1.00100000      1.00200100
    -1.00010000      1.00020001
> rightLim (g(x),-1,4):
    -0.90000000      0.81000000
    -0.99000000      0.98010000
    -0.99900000      0.99800100
    -0.99990000      0.99980001
```

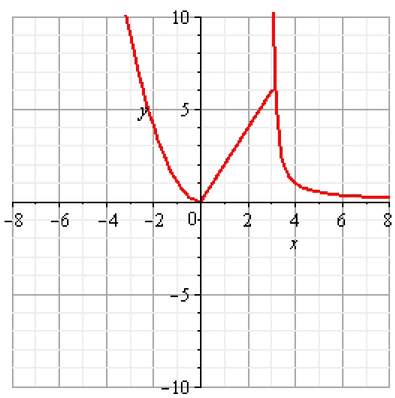
Support at x=1 and x=4 using `leftLim` and `rightLim`

```
> leftLim (g(x),1,4):
     0.90000000       0.81000000
     0.99000000       0.98010000
     0.99900000       0.99800100
     0.99990000       0.99980001
> rightLim (g(x),1,4):
     1.10000000       1.21000000
     1.01000000       1.02010000
     1.00100000       1.00200100
     1.00010000       1.00020001
> leftLim (g(x),4,4):
     3.90000000      15.21000000
     3.99000000      15.92010000
     3.99900000      15.99200100
     3.99990000      15.99920001
> rightLim (g(x),4,4):
     4.10000000      16.81000000
     4.01000000      16.08010000
     4.00100000      16.00800100
     4.00010000      16.00080001
```

Figure 121. Group P_3's analysis of mystery function g(x) in Lab 3

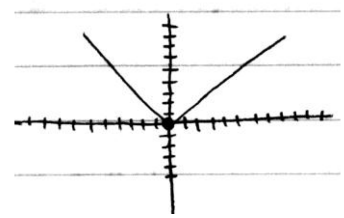
The analysis of function h (see Figure 122) was likewise problematic in that correct behavior was inferred at x=0 but the right behavior at x=3 was incorrect.

Actual Mystery Function $h(x)$

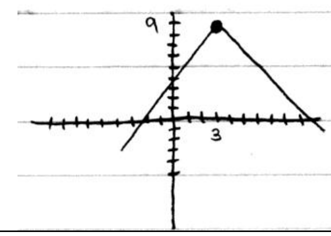


Corner at x=0 and a Vertical asymptote at x=3.

Inferred Behavior at x=0



Inferred Behavior at x=3



Support at x=0 using `leftLim` and `rightLim`

```
> h(x) := proc( x )
   if type(x, realcons) then
      x^2;
   else
      'fn(x)';
   end if:
end proc:
> leftLim (h(x),0,4):
    -0.10000000      0.01000000
    -0.01000000      0.00010000
    -0.00100000      0.00000100
    -0.00010000      0.00000001
> rightLim (h(x),0,4):
     0.10000000      0.01000000
     0.01000000      0.00010000
     0.00100000      0.00000100
     0.00010000      0.00000001
```

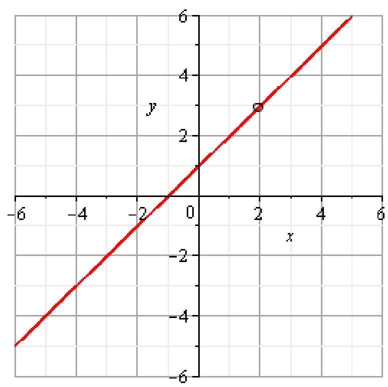
Support at x=3 using `leftLim` and `rightLim`

```
> leftLim (h(x),3,4):
     2.90000000      8.41000000
     2.99000000      8.94010000
     2.99900000      8.99400100
     2.99990000      8.99940001
> rightLim (h(x),3,4):
     3.10000000      9.61000000
     3.01000000      9.06010000
     3.00100000      9.00600100
     3.00010000      9.00060001
```

Figure 122. Group P_3's analysis of mystery function h(x) in Lab 3

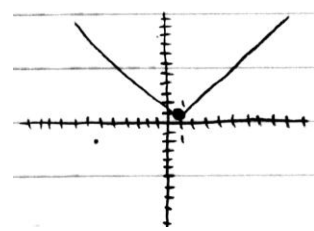
On function k, as on the previous functions, incorrect inferences were made on one side of the point of interest. These results are shown in Figure 123.

Actual Mystery Function $k(x)$

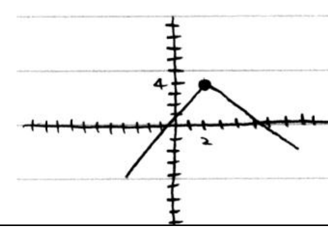


hole at x=2.

Inferred Behavior at x=1



Inferred Behavior at x=2



Support at x=1 using `leftLim` and `rightLim`

```
> k(x) := proc( x )
   if type(x, realcons) then
      x^2;
   else
      'k(x)';
   end if:
end proc:

> rightLim (k(x),1,4):
     1.10000000      1.21000000
     1.01000000      1.02010000
     1.00100000      1.00200100
     1.00010000      1.00020001
> leftLim (k(x),1,4):
     0.90000000      0.81000000
     0.99000000      0.98010000
     0.99900000      0.99800100
     0.99990000      0.99980001
```

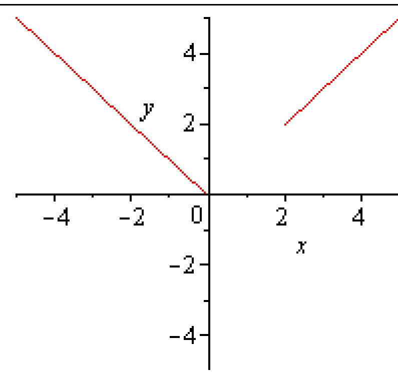
Support at x=2 using `leftLim` and `rightLim`

```
> rightLim (k(x),2,4):
     2.10000000      4.41000000
     2.01000000      4.04010000
     2.00100000      4.00400100
     2.00010000      4.00040001
> leftLim (k(x),2,4):
     1.90000000      3.61000000
     1.99000000      3.96010000
     1.99900000      3.99600100
     1.99990000      3.99960001
```

Figure 123. Group P_3's analysis of mystery function k(x) in Lab 3

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple with specified behavior. As shown in Figure 124, the pair was unable to create a hole at x=2 as well as unable to create the desired asymptotic behavior at x=-1 requested for function b.

Desired Behavior	**Function definition**	**Plot**

Hole at x= 2
Vertical Asymptote at x=-1
	`b := proc( x )`
	`if type(x,`
	`realcons) then`
	`if x>=2 then`
	`x;`
	`elif x<0 then`
	`-x;`
	`else`
	`undefined;`
	`end if:`
	`else`
	`'b(x)';`
	`end if:`
	`end proc:`
		

Supporting data produced with tools.	`> leftLim (b,2,4):`
	`1.900000 NaN`
	`1.990000 NaN`
	`1.999000 NaN`
	`1.999900 NaN`
	`> rightLim (b,2,4):`
	`2.100000 2.100000`
	`2.010000 2.010000`
	`2.001000 2.001000`
	`2.000100 2.000100`
		`> leftLim (b,-1,4):`
		`-1.100000 1.100000`
		`-1.010000 1.010000`
		`-1.001000 1.001000`
		`-1.000100 1.000100`
		`> rightLim (b,-1,4):`
		`-0.900000 0.900000`
		`-0.990000 0.990000`
		`-0.999000 0.999000`
		`-0.999900 0.999900`

Figure 124. Group P_3's construction of function b(x) in Lab 3

For function c (see Figure 125), the pair again produced neither the desired jump discontinuity nor the asymptotic behavior at x=-3. The demonstrated an ability to use the tool but did not explain how the resulting output supported their conclusions. Also, like group P_1, this group does not understand the final `else` clause is unnecessary- a programming related misunderstanding.

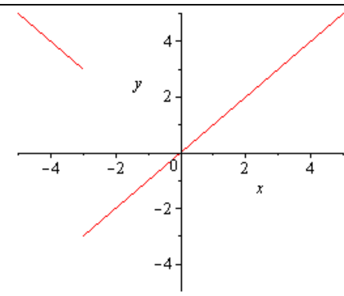
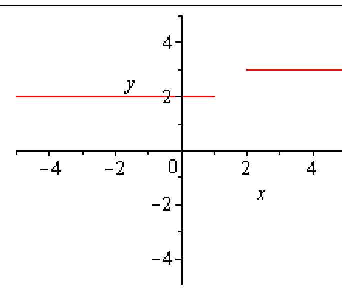
<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr><td>Jump discontinuity at x= -1
$\lim_{x\to -3^-} c(x) = +\infty$
$\lim_{x\to -3^+} c(x) = +\infty$</td><td><code>> c := proc(x)</code>
<code>if type(x, realcons) then</code>
<code>if x>-3 then</code>
<code>> x;</code>
<code>> elif x<-3 then</code>
<code>> -x;</code>
<code>> else</code>
<code>> undefined;</code>
<code>> end if:</code>
<code>> else</code>
<code>'c(x)';</code>
<code>end if:</code>
<code>> end proc:</code></td><td>
</td></tr>
<tr><td>Supporting data produced with tools.</td><td colspan="2"><code>> leftLim (c,-1,4):</code>
<code>-1.10000000 -1.10000000</code>
<code>-1.01000000 -1.01000000</code>
<code>-1.00100000 -1.00100000</code>
<code>-1.00010000 -1.00010000</code>
<code>> rightLim (c, -1,4):</code>
<code>-0.90000000 -0.90000000</code>
<code>-0.99000000 -0.99000000</code>
<code>-0.99900000 -0.99900000</code>
<code>-0.99990000 -0.99990000</code></td></tr>
</table>

Figure 125. Group P_3's construction of function c(x) in Lab 3

For function d (see Figure 126), the pair was able to accomplish the requested domain restriction using appropriate conditionals. Moreover, the function possessed appropriate limiting behavior at x=1 and x=2 as supported by their application of `leftLim` and `rightLim`.

Desired Behavior	**Function definition**	**Plot**

Undefined on [1,2]
$\lim_{x\to 1^-} d(x)=2$
$\lim_{x\to 2^+} d(x)=3$
	`> d := proc( x )`
	`   if type(x, realcons) then`
	`     if x>2 then`
	`>       3;`
	`>       elif x<1 then`
	`>       2;`
	`>       else`
	`>       undefined;`
	`>       end if:`
	`>      else`
	`      'd(x)';`
	`   end if:`
	`> end proc:`
		

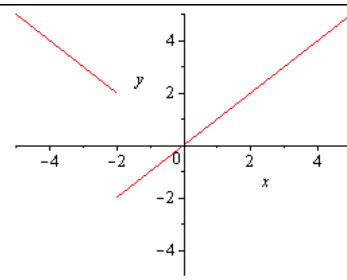
Supporting data produced with tools.	`> leftLim(d, 1,4):`
	`0.900000      2.000000`
	`0.990000      2.000000`
	`0.999000      2.000000`
	`0.999900      2.000000`
		`> rightLim( d, 2,4):`
		`2.100000      3.000000`
		`2.010000      3.000000`
		`2.001000      3.000000`
		`2.000100      3.000000`

Figure 126. Group P_3's construction of function d(x) in Lab 3

Function e satisfied only one of the criteria requested. As shown in Figure 127, the function possessed the correct value at x=1 but this fact was not supported by an appropriate evaluation. Additionally, the limiting behavior at x=1 and x=-2 was not accomplished nor was it properly justified with the computational tools.

The group's implementation of the function shown demonstrates confusion related to conditional statements in Maple. It further demonstrates confusion related to the construction of asymptotic behavior at a point. None of the component functions in the piecewise definition embody such behavior at any point. Even if the conditional statements had been properly understood, the requested asymptotic behavior would not have been implemented.

Desired Behavior	**Function definition**	**Plot**

$\lim_{x\to 1} e(x)=3$
$e(1)=5$
$\lim_{x\to -2^-} e(x)=+\infty$
$\lim_{x\to -2^+} e(x)=-\infty$
	`> e:= proc( x )`
	`  if type(x, realcons) then`
	`     if x=1 then`
	`        5;`
	`      elif x<-2 then`
	`        -x;`
	`      elif x>-2 then`
	`        x;`
	`      elif x>1 then`
	`        3;`
	`      else`
	`        undefined;`
	`    end if:`
	`    else`
	`      'e(x)';`
	` end if:`
		

Supporting data produced with tools.	`> leftLim (e,1,4):`
	`0.900000        0.900000`
	`0.990000        0.990000`
	`0.999000        0.999000`
	`0.999900        0.999900`
	`> rightLim (e,1,4):`
	`1.100000        1.100000`
	`1.010000        1.010000`
	`1.001000        1.001000`
	`1.000100        1.000100`
		`> leftLim (e,-2,4):`
		`-2.100000       2.100000`
		`-2.010000       2.010000`
		`-2.001000       2.001000`
		`-2.000100       2.000100`
		`> rightLim (e, -2,4):`
		`-1.900000      -1.900000`
		`-1.990000      -1.990000`
		`-1.999000      -1.999000`
		`-1.999900      -1.999900`

Figure 127. Group P_3's construction of function e(x) in Lab 3

Following the lab, the participating student completed a written post-lab activity to explore the degree to which she understood the coordination between the domain and range process of the limiting process. Two responses to questions on this activity demonstrate relevant interpretations offered by the student. These are shown in Figure 128.

From this student' conclusions, barring some inconsistent mathematical notation, she is very aware of the need for coordination between the domain and range processes and possess a good understanding of the limit process. While she appears to be able to effectively use the tool, she had difficulty with conditional expressions as well as difficulty constructing piecewise functions. Based upon her responses within the lab, the

Table 5		**Table 6**	
x	f(x)	x	f(x)
---------------	---------------	---------------	---------------
2.10000000	5.41000000	2.10000000	7.94010000
2.01000000	5.04010000	2.01000000	5.49230400
2.00100000	5.00400100	2.00100000	5.88537600
2.00010000	5.00040001	2.00010000	7.47392100
2.00001000	5.00004000	2.00001000	7.74976400
2.00000100	5.00000400	2.00000100	5.79644900
2.00000010	5.00000040	2.00000010	5.32522500
2.00000001	5.00000004	2.00000001	3.48168900
x	f(x)	x	f(x)
---------------	---------------	---------------	---------------
1.90000000	2.90000000	1.90000000	-2.72000000
1.99000000	2.99000000	1.99000000	-2.16200000
1.99900000	2.99900000	1.99900000	-1.09400000
1.99990000	2.99990000	1.99990000	-2.13800000
1.99999000	2.99999000	1.99999000	-2.28600000
1.99999900	2.99999900	1.99999900	-2.51000000
1.99999990	2.99999990	1.99999990	-2.68600000
1.99999999	2.99999999	1.99999999	-2.77000000
Conclusions:		Conclusions:	
The following limits exist. From top, $\underset{x\to 2}{f(x)}=5$ *and from the bottom* $\underset{x\to 2}{f(x)}=3$		*This table has no possible conclusions. Although the x-values in both tables approach 2, the y-values of f(x) are not parallel/consistent with the functions x-values..*	

Figure 128. Group P_3's response to Post-lab 3 questions

student demonstrated an understanding of the domain process, APOS 3a, as well as the need for a corresponding range process, APOS 3b. Additionally, an understanding of the coordinated relationship between these two processes is in evidence for this student, APOS step 3c. This student gave her peer a perfect peer review and stated "*I am starting to get a better understanding of how the program works. Although it is difficult at times, it is nice to be able to ask a partner a question and figure out what is wrong*."

Group P_4

The pair correctly constructed the two procedures, `leftLim` and `rightLim` as shown in Figure 129. Of note, the group utilized local variables, x, and i, suggesting an understanding of Maples' use of variables and scope- an understanding of an important programming construct. Additionally, the pair developed an appropriate looping procedure which provided the correct sequence of function evaluations. The specific loop suggests an interiorization of the domain process utilized in the limiting process, APOS step 2 and 3a.

Once these tools were developed, the pair utilized them to infer the behavior of the four mystery functions from the previous lab. As in Lab 2, the students did not have access to actual graphs of the mystery functions.

`leftLim` Procedure	**`rightLim` Procedure**

`> leftLim := proc( f, a, n)`
`local x, i;`
`x:=a-.01;`
`for i from 1 to n do`
` printf("%15.6f %15.6f\n",x,f(x));`
` x:=a-10^(-(i+1));`
`end do:`
`end proc:`
	`> rightLim := proc( f, a, n)`
	`local x, i;`
	`x:=a+.1;`
	`for i from 1 to n do`
	` printf("%15.6f %15.6f\n",x,f(x));`
	` x:=a+10^(-(i+1));`
	`end do:`
	`end proc:`

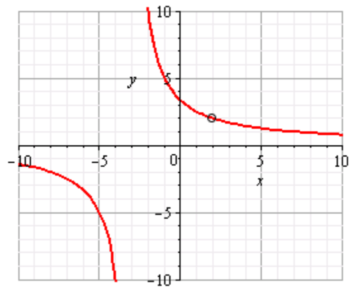
Figure 129. Group P_4's implementation of `leftLim` and `rightLim` procedures

Like groups P_1 and P_3, this pair attempted to redefine all the functions as the squaring function, see Figure 130, but unlike the other groups, they correctly used to same variable names as the mystery functions. As a result, their attempts to redefine the functions were unsuccessful and henceforward their analysis utilized the intended functions.

The pair did not provide sketches for any of the mystery functions. Shown below is the pair's first application of `leftLim` and `rightLim`. Clearly, the parameters to the procedures were not understood as is demonstrated in their application to function f (see Figure 131). The pair supplied both points x=-3 and x=2 as two parameters in a single call to the `leftLim` and `rightLim` procedures. This reflects a misunderstanding of the role of the parameters to the procedures. As the group did not provide a sketch, it is unclear what conclusion(s), if any, they drew from this output. Although not shown here, similar mistakes were made in the exploration of mystery functions g, h, and k resulting in no interpretations.

```
> f := proc( x )
   if type(x, realcons) then
      x^2;
   else
      'f(x)';
   end if:
end proc:

Error, attempting to assign to `MysteryFunctions:-f` which is
protected
```

Figure 130. Group P_4's attempt to redefine mystery function f(x) in Lab 3

<table>
<tr><th>Actual Mystery Function $f(x)$</th><th>Inferred Behavior</th></tr>
<tr><td>

Hole at x=2, Vertical asymptote at x= -3.</td><td>No sketch provided</td></tr>
<tr><td colspan="2">Support at x=2 and x=-3 using leftLim and rightLim

> leftLim (f, -3, 2):
-3.100000 -100.000000
-3.010000 -1000.000000
> rightLim (f, -3, 2):
-2.900000 100.000000
-2.990000 1000.000000</td></tr>
</table>

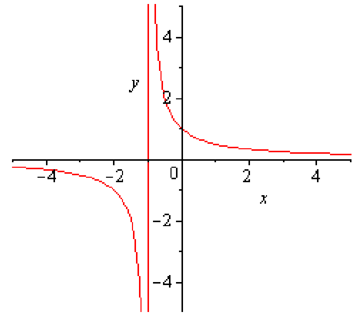
Figure 131. Group P_4's analysis of mystery function f(x) in Lab 3

As shown in Figure 133, the pair accomplished both the task of creating a hole at x=2 and creating a vertical asymptote at x=-1 as the resulting limit tables support. They made some progress demonstrating the hole at x=2 using the tools but failed to actually evaluate the function at x=2. No attempt was made to justify the asymptotic behavior at x=-1.

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple that have specified behavior. In this and subsequent work, the pair seems to have come to understand the procedure parameters to `leftLim` and `rightLim`.

With function c (see Figure 132), the pair correctly produced a function with appropriate asymptotic behavior at x=-3 and provided justification using the tools. The requested jump discontinuity at x=-1 was not supplied nor was it justified by the tool.

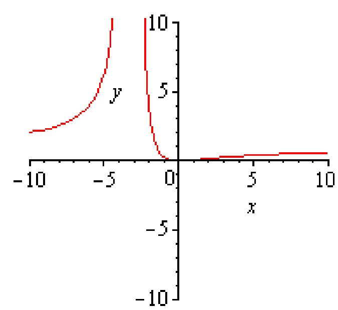
Desired Behavior	Function definition	Plot

Hole at x= 2
Vertical Asymptote at x=-1
	> b := proc(x)
	if type(x, realcons) then
	(x-2)/((x+1)*(x-2));
	else
	'b(x)';
	end if:
	end proc:
		

Supporting data produced with tools.	> leftLim(b, 2, 4): 1.900000 0.344828 1.990000 0.334448 1.999000 0.333444 1.999900 0.333344 > rightLim(b, 2, 4): 2.100000 0.322581 2.010000 0.332226 2.001000 0.333222 2.000100 0.333322	

Figure 133. Group P_4's construction of function b(x) in Lab 3

Desired Behavior	Function definition	Plot

Jump discontinuity at x= -1
$\lim_{x \to -3^-} c(x) = +\infty$
$\lim_{x \to -3^+} c(x) = +\infty$
	> c := proc(x)
	if type(x, realcons) then
	(x/(x+3))^2;
	else
	'c(x)';
	end if:
	end proc:
		

Supporting data produced with tools.	> leftLim(c, -3, 3): -3.100000 961.000000 -3.010000 90601.000000 -3.001000 9006001.000000 > rightLim(c, -3, 3): -2.900000 841.000000 -2.990000 89401.000000 -2.999000 8994001.000000	

Figure 132. Group P_4's construction of function c(x) in Lab 3

Function d, shown in Figure 134, expressed the requested domain restriction on [1,2]. Appropriate one sided behavior was present to the left of x=1. The right side behavior at x=2 would have been correct if the pair had not reversed the inequality sign on the conditional expression, i.e. `elif x >=2` rather than `x <= 2`. No supporting data was provided using either tool.

<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr><td>Undefined on [1,2]
$\lim_{x\to 1^-} d(x) = 2$
$\lim_{x\to 2^+} d(x) = 3$</td><td>

```
d := proc (x)
  if type(x, realcons) then
    if 1 <= x and x <= 2 then
      undefined
    elif x <= 1 then
      2
    elif x <= 2 then
      3
    end if
  else
    'd(x)'
  end if
 end proc:
```

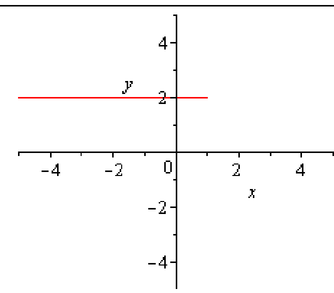
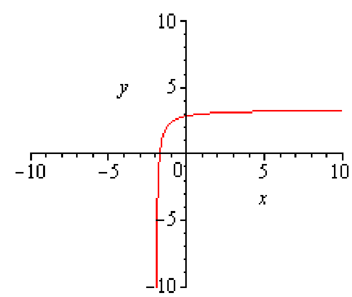
</td><td>
</td></tr>
<tr><td>Supporting data produced with tools.</td><td>No support provided.</td><td></td></tr>
</table>

Figure 134. Group P_4's construction of function d(x) in Lab 3

Function e (see Figure 135) satisfied two of the four criteria specified; the graph possessed the correct value at x=1 as demonstrated using the tool. The asymptotic behavior from the right but not the left of x=-2 was also supported. However, the pair created a horizontal asymptote y=10/3. This was perhaps an unsuccessful attempt to satisfy the requirement that $\lim_{x\to 1} e(x) = 3$ having confused the definition of horizontal and vertical asymptote. No attempt was made justify this behavior using the tool.

<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr><td>$\lim_{x\to 1} e(x) = 3$
$e(1) = 5$
$\lim_{x\to -2^-} e(x) = +\infty$
$\lim_{x\to -2^+} e(x) = -\infty$</td><td>

```
> e := proc(x)
if type (x, realcons) then
   if x=1 then
      5;
   elif x>-2 then
      -(1/(x+2))+(10/3);
   else
      undefined;
   end if:
else
   'e(x)';
end if:
end proc:
```

</td><td>
</td></tr>
<tr><td>Supporting data produced with tools.</td><td colspan="2">

```
> e(1);
      5
> rightLim(e, -2, 3):
-1.900000        -6.666667
-1.990000       -96.666667
-1.999000      -996.666667
> leftLim(e, -2, 3):
-2.100000              NaN
-2.010000              NaN
-2.001000              NaN
```

</td></tr>
</table>

Figure 135. Group P_4's construction of function d(x) in Lab 3

Following the lab, the student completed a written post-lab activity to explore the degree to which he understood the coordination between the domain and range process of the limiting process. Instructive sample responses to these questions appear in Figure 136.

Post-lab 3: Sample Responses

Table 2		**Table 4**	
x	f(x)	x	f(x)
---------------	---------------	---------------	---------------
2.10000000	4.87930340	2.10000000	16.11758758
2.01000000	4.08722195	2.01000000	12.50332216
2.00100000	4.00871339	2.00100000	10.07769600
2.00010000	4.00087125	2.00010000	17.57600000
2.00001000	4.00008712	2.00001000	8.21794983
2.00000100	4.00000871	2.00000100	16.71830269
2.00000010	4.00000087	2.00000010	13.16097188
2.00000001	4.00000009	2.00000001	9.48773561
x	f(x)	x	f(x)
---------------	---------------	---------------	---------------
1.68000000	6.85900000	1.90000000	6.85900000
1.87800000	7.88059900	1.99000000	7.88059900
1.62000000	7.98800600	1.99900000	7.98800600
1.16200000	7.99880006	1.99990000	7.99880006
1.40500000	7.99988000	1.99999000	7.99988000
1.71600000	7.99998800	1.99999900	7.99998800
1.70100000	7.99999880	1.99999990	7.99999880
1.29000000	7.99999988	1.99999999	7.99999988
Conclusions:		Conclusions:	
$\lim_{x \to 2^+} f(x) = 4$ and $\lim_{x \to 2^-} f(x) = 8$		$\lim_{x \to 2^+} f(x) = \infty$ and $\lim_{x \to 2^-} f(x) = 8$	

Figure 136. Group P_4's Selected Responses to Post-lab 3

From these responses it is clear that the student does not have an awareness of the requirement of coordination between the domain and range processes. In the table on the left, the student concludes the limit is 8. Clearly he is focusing primarily on the range process. When the range process is not convergent, he concludes the limit is infinite. Apparently, when the domain process fails to converge, the limit does not exist, and when the range process does not converge, the limit is infinite.

The student initially had difficulty using the tool but eventually came to understand its operation. However, the student did not effectively utilize the tool to support his answers. Based upon there inferences within the lab itself, the pair demonstrated an understanding of the domain process, APOS 3a, and possessed an understanding of the range process, APOS 3b. However, an understanding of the coordination of these two processes, APOS step 3c, was not in evidence. This student indicated that he was glad he was "*glad you paired people you knew were friends*."

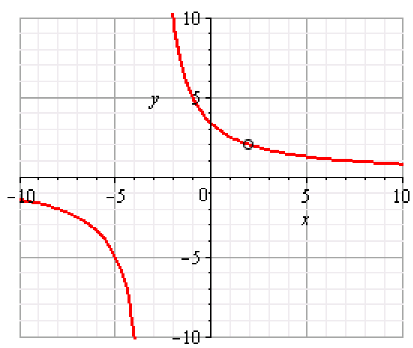
Group N_1

Using the `leftLim` and `rightLim` tools, the pair utilized them to infer the behavior of the four mystery functions from Lab 2. The mystery graphs are shown for comparative purposes. The group's analysis of mystery function f is shown in Figure 137.

The pair inferred a vertical asymptote at x=-3 using the resulting tables shown. No analysis was done to explore the graph near x=2. Unfortunately, the way in which they call the `leftLim` procedure (see Figure 137) indicates the group misunderstands the function of the parameters to the procedure. This is further reinforced by their subsequent analysis of function g, see Figure 138. Specifically, when the pair was asked to analyze the graph at x=-3 and x=2, the group provided *both* points as parameters to `leftLim`

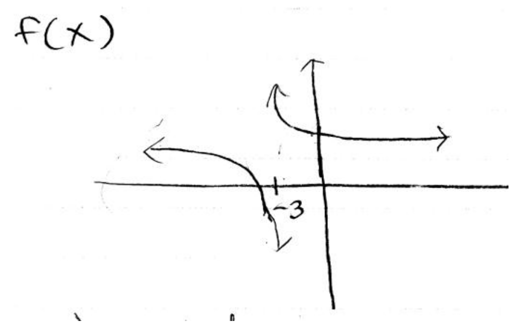
and `rightLim` rather than a single point and a specification of the degree of closeness desired. The pair does not understand that the second parameter is the point and the third parameter is an integer specifying the degree of closeness to that point. Thus, luck played a significant role in their accurate inference of f's behavior around x=-3. In spite of misinterpreting the parameters functions, they tables produced provided relevant data relating to graph f at x=-3 which did lead to an appropriate inference. Subsequently, it is unclear why they evaluated the function at x=0.

Actual Mystery Function $f(x)$



Hole at x=2, Vertical asymptote at x= -3.

Inferred Behavior



Support at x=-3 and x=2 using `leftLim` and `rightLim`

```
> leftLim(f, -3, 2);
[Left Limit]   x              f(x)
--------------- ---------------
    -3.10000000   -100.00000000
    -3.01000000  -1000.00000000
> f(0);
```

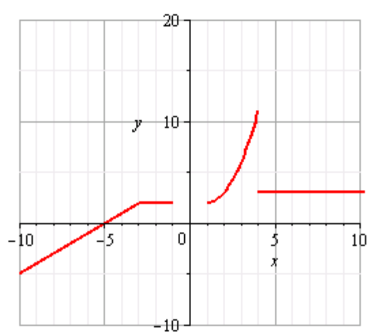
3.33333333

```
> rightLim(f,-3,2);
[Right Limit] x               f(x)
--------------- ---------------
    -2.90000000    100.00000000
    -2.99000000   1000.00000000
```

Figure 137. Group N_1's analysis of mystery function f(x) in Lab 3

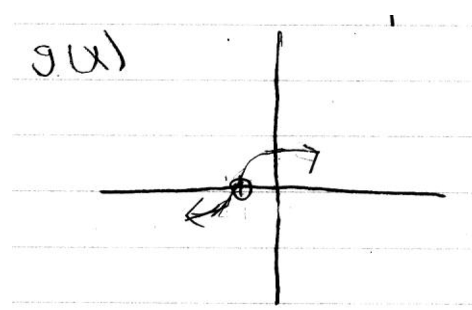
They were similarly unsuccessful in their analysis of g, h, and k as shown in Figure 138, Figure ***139***, and Figure ***140***. Their graph of g, shown in Figure 138, in no way resembles the actual graph of g. Again, not understanding the parameters to the `leftLim` and `rightLim` procedures lead to erroneous interpretations of the graph's behavior. As with function, f, the pair used all three points of interest as parameters to the procedures. Figure 139

Actual Mystery Function $g(x)$



Jump at x=4, no left limit at x=1, no right limit at x=-1.

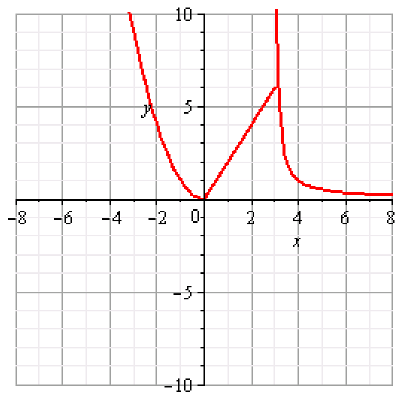
Inferred Behavior



Support at x=-1, 1, and 4 using `leftLim` and `rightLim`

```
> leftLim(g, -1, 1, 4);
[Left Limit]  x              f(x)
--------------- ---------------
    -1.10000000      2.00000000
> g(0);
    Undefined
> rightLim(g, -1, 1, 4);
[Right Limit] x              f(x)
--------------- ---------------
     -.90000000             NaN
```

Figure 138. Group N_1's analysis of mystery function g(x) in Lab 3

<table>
<tr><th>Actual Mystery Function $h(x)$</th><th>Inferred Behavior</th></tr>
<tr><td>

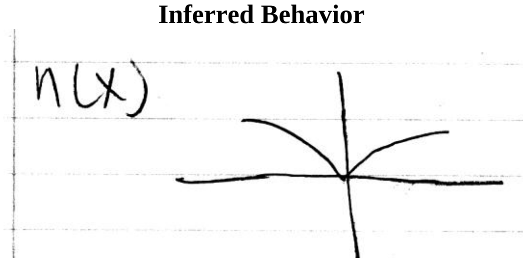
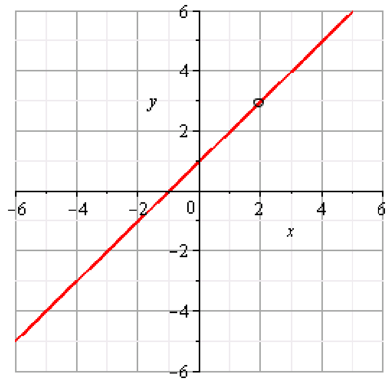
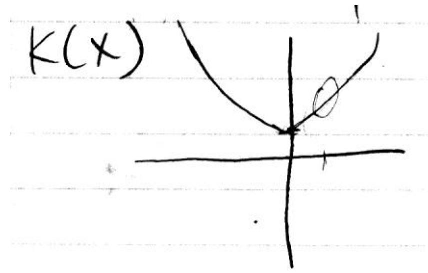
Corner at x=0 and a Vertical asymptote at x=3.</td><td>
</td></tr>
<tr><td>Support at x=0 using <code>leftLim</code> and <code>rightLim</code>
<code>> leftLim(h, 0, 3);</code>
<code>[Left Limit] x f(x)</code>
<code>--------------- ---------------</code>
<code>-0.10000000 0.01000000</code>
<code>-0.01000000 0.00010000</code>
<code>-0.00100000 0.00000100</code>
<code>-0.00001000 0.00002000</code>
<code>-0.00000100 0.00000200</code></td><td>Support at x=3 using <code>leftLim</code> and <code>rightLim</code>
<code>> rightLim(h,0,6);</code>
<code>[Right Limit] x f(x)</code>
<code>--------------- ---------------</code>
<code>.10000000 .20000000</code>
<code>.01000000 .02000000</code>
<code>.00100000 .00200000</code>
<code>.00010000 .00020000</code>
<code>> h(0);</code>
0</td></tr>
</table>

Figure 139. Group N_1's analysis of mystery function h(x) in Lab 3

<table>
<tr><th>Actual Mystery Function $k(x)$</th><th>Inferred Behavior</th></tr>
<tr><td>

hole at x=2.</td><td>
</td></tr>
</table>

Support at x=1 and x=2 using `leftLim` and `rightLim`

```
> leftLim(k, 1, 2);
[Left Limit]   x            f(x)
--------------- ---------------
     0.90000000      1.90000000
     0.99000000      1.99000000
> rightLim(k,1,2);
[Right Limit] x             f(x)
--------------- ---------------
     1.10000000      2.10000000
     1.01000000      2.01000000
> k(0);
1
```

Figure 140. Group N_1's analysis of mystery function k(x) in Lab 3

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple that have specified behavior.

As shown in Figure 141, the pair accomplished the task of creating function b possessing a hole at x=2 and a vertical asymptote at x=-1. They were successful in creating a suitable function in Maple but provided no justification using the `leftLim` and `rightLim` tools for the requested behavior nor did they use evaluation to verify the function possessed a hole at x=2. This is not surprising as the group clearly does not understand how to utilize the tool.

They explained how they created the requested function as follows. "*Since there is a hole at x=2, the numerator & denominator must include (x-2) and the vertical*

asymptote (x+1) is on the denominator." The pair used methods explored in a prior math course for producing such behavior but could not use the `leftLim` and `rightLim` tools to justify the behavior. Additionally, the phrase *"...and the vertical asymptote (x+1) is in the denominato*r" suggests that don't have a clear understanding of an asymptote; they fail to differentiate between the asymptote, and the factor, x+1, that causes the asymptotic behavior.

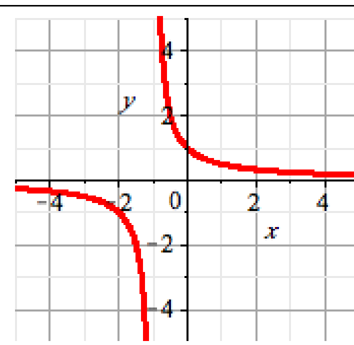
Desired Behavior	**`Function definition`**	**Plot**
Hole at x= 2 Vertical Asymptote at x=-1	`b:= x-> ((x-2)/(x^2-x-2));`	
Supporting data produced with tools.	None	

Figure 141. Group N_1's construction of function b(x) in Lab 3

Again, for function c (see Figure 142), the pair produced none of the requested behavior. They were unable to effectively construct an appropriate procedure and make accurate inferences using the data produced with the tools. The pair was able to create a discontinuity but not at the requested location, x=-1 and they did not produce any asymptotic behavior.

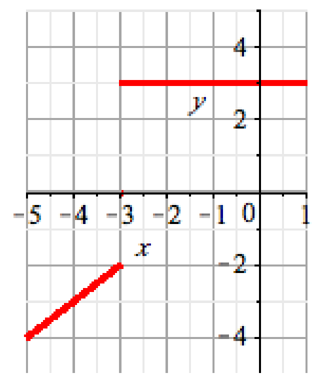
<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr><td>Jump discontinuity at x= -1
$\lim_{x \to -3^-} c(x) = +\infty$
$\lim_{x \to -3^+} c(x) = +\infty$</td><td><pre>c:=x-> piecewise(
    x > -3, 3,
    x < -3, x+1);</pre></td><td>
</td></tr>
<tr><td>Supporting data produced with tools.</td><td colspan="2">None</td></tr>
</table>

Figure 142. Group N_1's construction of function c(x) in Lab 3

Function d, shown in Figure 143, does possess the requested domain restriction as well as the requested limiting behavior. As with the previous functions, the pair does not provide any justification using `leftLim` and `rightLim` tools.

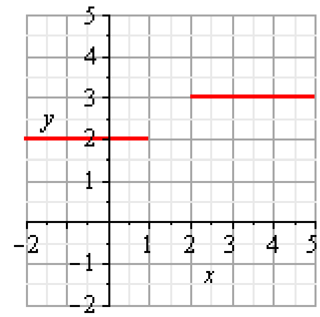
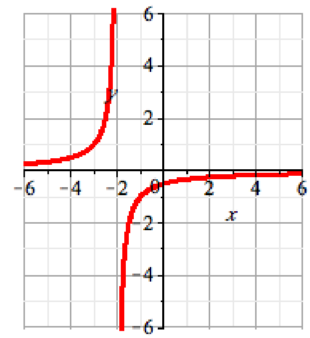
<table>
<tr><th>Desired Behavior</th><th><code>Function definition</code></th><th>Plot</th></tr>
<tr><td>Undefined on [1,2]
$\lim_{x\to 1^-} d(x) = 2$
$\lim_{x\to 2^+} d(x) = 3$</td><td><code>d:=x-> piecewise(</code>
<code>x<1, 2,</code>
<code>x>2,3,</code>
<code>undefined);</code></td><td>
</td></tr>
<tr><td>Supporting data produced with tools.</td><td colspan="2">None</td></tr>
</table>

Figure 143. Group N_1's construction of function d(x) in Lab 3

Function e (see Figure 144) satisfied all the criteria specified and was not supported with the application of `leftLim` and `rightLim` and evaluation.

Desired Behavior	**Function definition**	**Plot**

$\lim\limits_{x\to 1} e(x) = 3$
$e(1) = 5$
$\lim\limits_{x\to -2^-} e(x) = +\infty$
$\lim\limits_{x\to -2^+} e(x) = -\infty$
	`e:=x->piecewise(`
	`x=1, 5,`
	`x>-2, (-1/(x+2)),`
	`x<-2,-1/(x+2),`
	`undefined);`
		

Supporting data produced with tools.	None	

Figure 144. Group N_1's construction of function e(x) in Lab 3

Following the lab, the group completed a written post-lab activity to explore the degree to which they understood the coordination between the domain and range process of the limiting process. The two group members gave responses that differed significantly and are shown in Figure 145.

Post-lab 3: Sample Responses

Table 2

x	f(x)
2.10000000	4.87930340
2.01000000	4.08722195
2.00100000	4.00871339
2.00010000	4.00087125
2.00001000	4.00008712
2.00000100	4.00000871
2.00000010	4.00000087
2.00000001	4.00000009

x	f(x)
1.68000000	6.85900000
1.87800000	7.88059900
1.62000000	7.98800600
1.16200000	7.99880006
1.40500000	7.99988000
1.71600000	7.99998800
1.70100000	7.99999880
1.29000000	7.99999988

Conclusions:

As the limit of the function is 2 from the right, the function gets closer to 4. As the limit of the function is 2 from the left, the function gets closer to 8.

Table 4

x	f(x)
2.10000000	16.11758758
2.01000000	12.50332216
2.00100000	10.07769600
2.00010000	17.57600000
2.00001000	8.21794983
2.00000100	16.71830269
2.00000010	13.16097188
2.00000001	9.48773561

x	f(x)
1.90000000	6.85900000
1.99000000	7.88059900
1.99900000	7.98800600
1.99990000	7.99880006
1.99999000	7.99988000
1.99999900	7.99998800
1.99999990	7.99999880
1.99999999	7.99999988

Conclusions:

When x approaches 2 from the right there is a jump discontinuity and as it further approaches to the left there is continuity and increases to 8.

Student B

Conclusions:

The first chart tells you that as you get closer and closer to x=2, the y-value gets closer and closer to 4, therefore $\lim_{x\to 2^+} f(2) = 4$. *You can conclude nothing from the second table.*

Conclusions:

Nothing can be concluded from the first table. As the x-values approach 2 from the left, the limit is 8. Therefore the left sided limit is 8.

Figure 145. Group N_1's Selected Responses to Post-lab 3

Student B demonstrated an awareness of the need for a coordinated domain and range process in the limiting process whereas student A did not. The group was unable to utilize `leftLim` and `rightLim` to justify any of their responses. One of the two clearly had not conceptualized the domain-range coordination, APOS step 3c. In fact, one student was unable to infer that a limit failed to exist when either the domain or range process failed to converge. Specifically, they did not achieve an understanding of APOS step 3b. This group experienced difficulties stemming from two main sources.

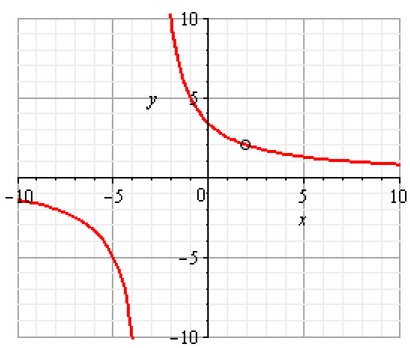
First, due the programming related issue of not understanding parameters to procedures, the pair could not make use of either of the tools. Therefore, as a group, the pair did not progress beyond APOS step 3a. Arguably there are still some issues related to step 3a as well.

The group was quite aware of their difficulties as expressed in their peer reviews. Each gave the other a perfect review and made the following comments "*This lab was a little confusing to us, but having a partner helps*." And "*We had trouble graphing many of the graphs because of missing simple instructions but my partner cooperated with me in completing the lab.*"

Group N_2

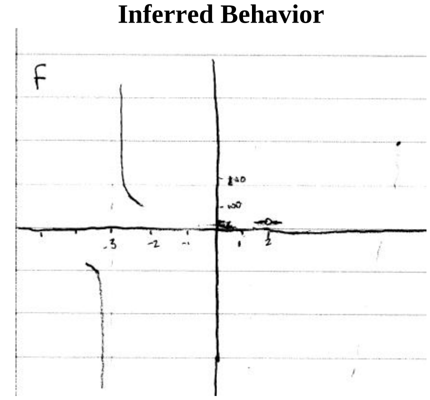
Using the `leftLim` and `rightLim` tools, the pair inferred the behavior of the four mystery functions. In analyzing mystery function f, see Figure 146, the pair accurately inferred a vertical asymptote at x=-3 using the resulting tables shown. Additionally, they additionally inferred the presence of a hole at x=2 using `leftLim`, `rightLim`, and evaluation at x=2.

Actual Mystery Function $f(x)$



Hole at x=2, Vertical asymptote at x= -3.

Inferred Behavior



Support at x=-3 and x=2 using `leftLim` and `rightLim`

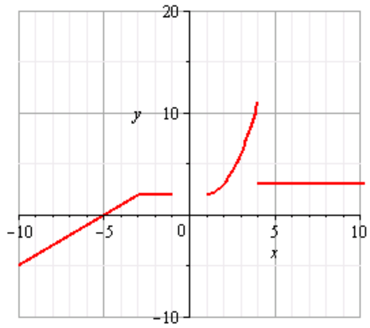
```
> leftLim(f,-3,4);
[Left Limit]   x              f(x)
---------------- ----------------
    -3.10000000    -100.00000000
    -3.01000000   -1000.00000000
    -3.00100000  -10000.00000000
    -3.00010000 -100000.00000000
> rightLim(f,-3,4)
[Left Limit]   x              f(x)
---------------- ----------------
    -3.10000000    -100.00000000
    -3.01000000   -1000.00000000
    -3.00100000  -10000.00000000
    -3.00010000 -100000.00000000
```

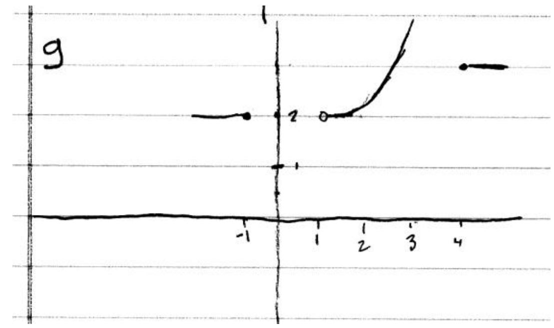
```
> leftLim(f,2,4);
rightLim(f,2,4);
[Left Limit]   x              f(x)
---------------- ----------------
     1.90000000       2.04081633
     1.99000000       2.00400802
     1.99900000       2.00040008
     1.99990000       2.00004000
[Right Limit]  x              f(x)
---------------- ----------------
     2.10000000       1.96078431
     2.01000000       1.99600798
     2.00100000       1.99960008
     2.00010000       1.99996000
> f(2);
      undefined
```

Figure 146. Group N_2's analysis of function f(x) in Lab 3

This response demonstrates a clear ability to interpret limiting behavior as well an understanding of the concept of asymptote and hole. Of particular interest is that this pair only shows behavior near the indicated point suggesting an understanding that such limiting behavior can only yield local information near respective points.

The partners have similar success in their analysis of functions g, h, and k. In each analysis of the respective mystery function, shown in Figure 147, Figure 148, and Figure 149, the pair accurately inferred the behavior of the function; their proposed graphs closely resemble the actual graphs.

<table>
<tr><th>Actual Mystery Function $g(x)$</th><th>Inferred Behavior</th></tr>
<tr><td>


Jump at x=4, no left limit at x=1, no right limit at x=-1.
</td><td>


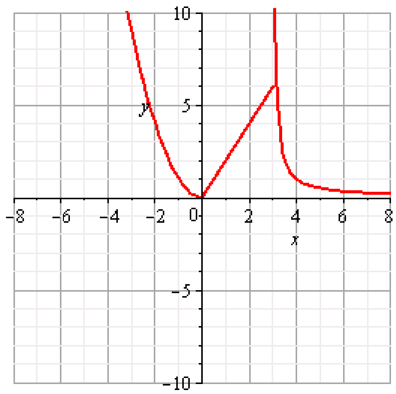
</td></tr>
<tr><td colspan="2">Support at x=-1, 1, and 4 using <code>leftLim</code> and <code>rightLim</code></td></tr>
</table>

```
> leftLim(g,-1,4);
  rightLim(g,-1,4);
  g(-1);
[Left Limit]  x              f(x)
--------------- ---------------
    -1.10000000      2.00000000
    -1.01000000      2.00000000
    -1.00100000      2.00000000
    -1.00010000      2.00000000
[Right Limit] x              f(x)
--------------- ---------------
    -0.90000000             NaN
    -0.99000000             NaN
    -0.99900000             NaN
    -0.99990000             NaN

2
```

```
>leftLim(g,1,4);rightLim(g,1,4);
g(1);
[Left Limit]  x              f(x)
--------------- ---------------
     0.90000000             NaN
     0.99000000             NaN
     0.99900000             NaN
     0.99990000             NaN
[Right Limit] x              f(x)
--------------- ---------------
     1.10000000      2.01000000
     1.01000000      2.00010000
     1.00100000      2.00000100
     1.00010000      2.00000001
undefined
>
leftLim(g,4,4);rightLim(g,4,4);g(4)
[Left Limit]  x              f(x)
--------------- ---------------
     3.90000000     10.41000000
     3.99000000     10.94010000
     3.99900000     10.99400100
     3.99990000     10.99940001
[Right Limit] x              f(x)
--------------- ---------------
     4.10000000      3.00000000
     4.01000000      3.00000000
     4.00100000      3.00000000
     4.00010000      3.00000000
11
```

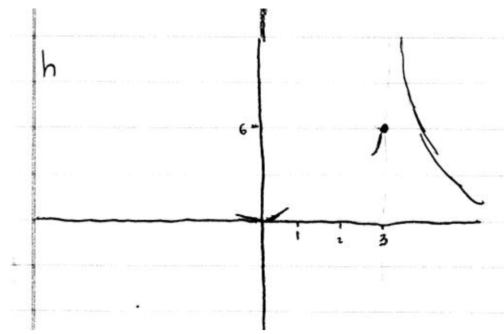
Figure 147. Group N_2's analysis of mystery function g(x) in Lab 3

Actual Mystery Function $h(x)$



Corner at x=0 and a Vertical asymptote at x=3.

Inferred Behavior



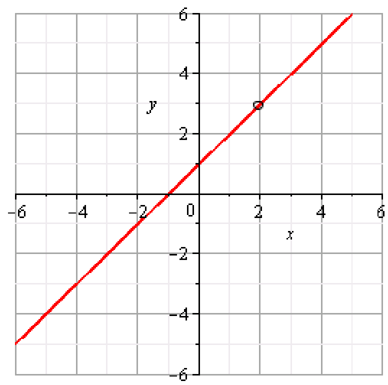
Support at x=0 and x=3 using `leftLim` and `rightLim`

```
> leftLim(h,0,4); rightLim(h,0,4);
h(0);

[Left Limit]  x              f(x)
--------------- ---------------
    -0.10000000      0.01000000
    -0.01000000      0.00010000
    -0.00100000      0.00000100
    -0.00010000      0.00000001
[Right Limit] x              f(x)
--------------- ---------------
     0.10000000      0.20000000
     0.01000000      0.02000000
     0.00100000      0.00200000
     0.00010000      0.00020000

0
```

```
> leftLim(h,3,4);
rightLim(h,3,4); h(3);

[Left Limit]  x              f(x)
--------------- ---------------
     2.90000000      5.80000000
     2.99000000      5.98000000
     2.99900000      5.99800000
     2.99990000      5.99980000
[Right Limit] x              f(x)
--------------- ---------------
     3.10000000     10.00000000
     3.01000000    100.00000000
     3.00100000   1000.00000000
     3.00010000  10000.00000000

6
```

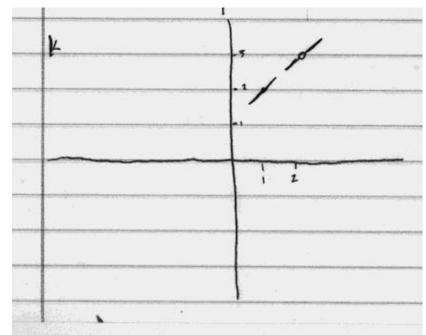
Figure 148. Group N_2 's analysis of mystery function h(x) in Lab 3

Actual Mystery Function $k(x)$



hole at x=2.

Inferred Behavior



Support at x=1 and x=2 using `leftLim` and `rightLim`

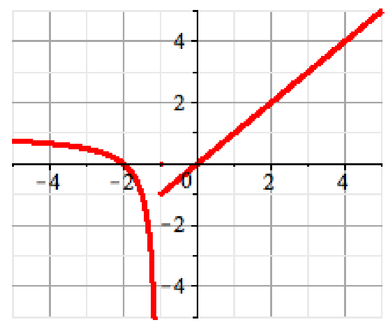
```
> leftLim(k,1,4); rightLim(k,1,4);
k(1);
[Left Limit]   x             f(x)
---------------  ---------------
     0.90000000       1.90000000
     0.99000000       1.99000000
     0.99900000       1.99900000
     0.99990000       1.99990000
[Right Limit]  x             f(x)
---------------  ---------------
     1.10000000       2.10000000
     1.01000000       2.01000000
     1.00100000       2.00100000
     1.00010000       2.00010000

2
```

```
> leftLim(k,2,4);
rightLim(k,2,4); k(2);
[Left Limit]   x             f(x)
---------------  ---------------
     1.90000000       2.90000000
     1.99000000       2.99000000
     1.99900000       2.99900000
     1.99990000       2.99990000
[Right Limit]  x             f(x)
---------------  ---------------
     2.10000000       3.10000000
     2.01000000       3.01000000
     2.00100000       3.00100000
     2.00010000       3.00010000

undefined
```

Figure 149. Group N_2 's analysis of mystery function k(x) in Lab 3

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple that have specified behavior. As shown in Figure 150, the pair accomplished the task of creating function b possessing a hole at x=2 and a vertical asymptote at x=-1. They were also successful in providing clear justification using the `leftLim` and `rightLim` tools.

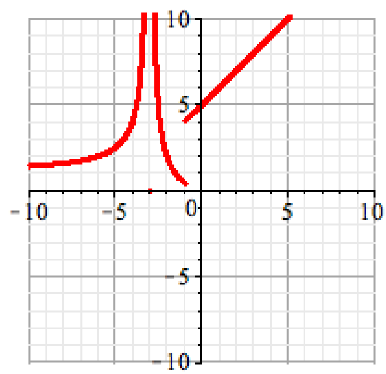
Desired Behavior	**Function definition**	**Plot**

Hole at x= 2
Vertical Asymptote at x=-1
	`b := x->piecewise(`
	`x<-1,   (x+2)/(x+1),`
	`x=2,    undefined,`
	`x>-1,   x);`
		

Supporting data produced with tools.	`> leftLim(b,2,4);`
	`rightLim(b,2,4); b(2);`
	`[Left Limit]   x        f(x)`
	`---------------- -----------`
	`1.900000   1.900000`
	`1.990000   1.990000`
	`1.999000   1.999000`
	`1.999900   1.999900`
	`[Right Limit] x        f(x)`
	`---------------- -----------`
	`2.100000   2.100000`
	`2.010000   2.010000`
	`2.001000   2.001000`
	`2.000100   2.000100`
	
	`undefined`
		`> leftLim(b,-1,4);`
		`rightLim(b,-1,4); b(-1);`
		`[Left Limit]  x    f(x)`
		`------------ ------------`
		`-1.100000   -9.000000`
		`-1.010000   -99.000000`
		`-1.001000   -999.000000`
		`-1.000100   -9999.000000`
		`[Right Limit]  x    f(x)`
		`------------ ---------`
		`-0.900000   -0.900000`
		`-0.990000   -0.990000`
		`-0.999000   -0.999000`
		`-0.9999000   -0.999900`
		
		`0`

Figure 150. Group N_2 's construction of function b(x) in Lab 3

For function c, the pair produced all of the requested behavior through effective construction and appropriate procedure use. Accurate inferences regarding the functions limiting behavior were made utilizing data produced with the tools (see Figure 151). Notably, the pair was observed using the tool to incrementally test limiting behavior as they proposed candidate functions. Rather than work on paper initially, the group proposed candidate functions in Maple and then tested that behavior utilizing the `leftLim` and `rightLim` tools; there was exploration and not just verification with the tool in which the tool was used to guide their construction.

<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr><td>Jump discontinuity at x= -1
$\lim_{x\to -3^-} c(x) = +\infty$
$\lim_{x\to -3^+} c(x) = +\infty$</td><td>c:=x->piecewise(
x<-3, (x)/(x+3),
x>-3 and x<=-1,-x/(x+3),
x>-1, x+5);</td><td></td></tr>
<tr><td>Supporting data produced with tools.</td><td>> leftLim(c,-1,4);
rightLim(c,-1,4);
c(-1);
[Left Limit] x f(x)
---------------- ---------
-1.100000 0.578947
-1.010000 0.507537
-1.001000 0.500750
-1.000100 0.500075
[Right Limit] x f(x)
---------------- ---------
-0.900000 4.100000
-0.990000 4.010000
-0.999000 4.001000
-0.999900 4.000100

1/2</td><td>> leftLim(c,-3,4);
rightLim(c,-3,4);
c(-3);
[Left Limit] x (x)
---------------- ---------
-3.100000 31.00000
-3.010000 301.00000
-3.001000 3001.00000
-3.000100 30001.00000
[Right Limit] x f(x)
---------------- ---------
-2.900000 29.00000
-2.990000 299.00000
-2.999000 2999.00000
-2.999900 29999.00000

0</td></tr>
</table>

Figure 151. Group N_2 's construction of function c(x) in Lab 3

For function d (see Figure 152), the pair was not able to accomplish the requested domain restriction but was able to construct the requested limiting behavior. As with the previous functions, the pair provided appropriate justification using `leftLim` and `rightLim` tools.

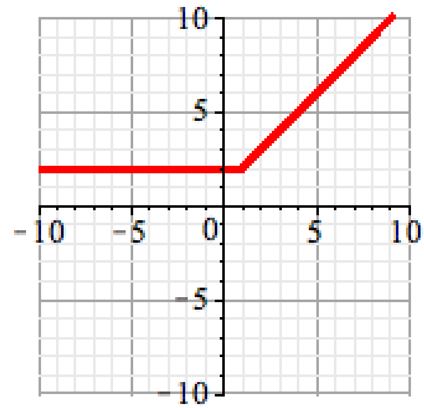
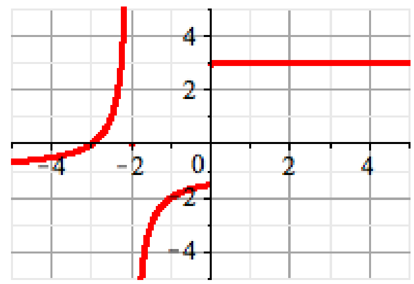
Desired Behavior	**Function definition**	**Plot**
Undefined on [1,2] $\lim_{x\to 1^-} d(x) = 2$ $\lim_{x\to 2^+} d(x) = 3$	d :=x->piecewise(x=1, undefined, x>1, x+1, x<1, 2);	
Supporting data produced with tools.	> leftLim(d,1,4); rightLim(d,1,4); d(1); [Left Limit] x f(x) -------------- -------- 0.900000 2.000000 0.990000 2.000000 0.999000 2.000000 0.999900 2.000000 [Right Limit] x f(x) --------------- ------- 1.100000 2.100000 1.010000 2.010000 1.001000 2.001000 1.000100 2.000100 undefined	> leftLim(d,2,4); rightLim(d,2,4); d(2); [Left Limit] x f(x) --------------- -------------- 1.90000000 2.90000000 1.99000000 2.99000000 1.99900000 2.99900000 1.99990000 2.99990000 [Right Limit] x f(x) --------------- -------------- 2.10000000 3.10000000 2.01000000 3.01000000 2.00100000 3.00100000 2.00010000 3.00010000 3

Figure 152. Group N_2's construction of function d(x) in Lab 3

Function e satisfied all the criteria specified and was supported with the application of `leftLim` and `rightLim` and evaluation (see Figure 153).

Desired Behavior	**Function definition**	**Plot**

$\lim_{x\to 1} e(x)=3$
$e(1)=5$
$\lim_{x\to -2^-} e(x)=+\infty$
$\lim_{x\to -2^+} e(x)=-\infty$
	`e := x->piecewise(`
	`x<-2, -(x+3)/(x+2),`
	`x>-2 and x < 0,`
	`-(x+3)/(x+2),`
	`x>=0 and x<1, 3,`
	`x=1, 5,`
	`x>1, 3);`
		

Supporting data produced with tools.	`> leftLim(e,1,4);`
	`rightLim(e,1,4); e(1);`
	`[Left Limit]  x   f(x)`
	`-------------- ---------`
	`0.900000       3.000000`
	`0.990000       3.000000`
	`0.999000       3.000000`
	`0.999900       3.000000`
	`[Right Limit] x   f(x)`
	`-------------- ---------`
	`1.100000       3.000000`
	`1.010000       3.000000`
	`1.001000       3.000000`
	`1.000100       3.000000`
	
	`5`
		`> leftLim(e,-2,4);`
		`rightLim(e,-2,4);`
		`[Left Limit]  x   f(x)`
		`---------- ---------------`
		`-2.100000       9.000000`
		`-2.010000      99.000000`
		`-2.001000     999.000000`
		`-2.000100    9999.000000`
		`[Right Limit] x   f(x)`
		`---------- ---------------`
		`-1.900000     -11.000000`
		`-1.990000    -101.000000`
		`-1.999000   -1001.000000`
		`-1.999900 -10001.000000`

Figure 153. Group N_2's construction of function e(x) in Lab 3

Following the lab, the group completed a written post-lab activity to explore the degree to which they understood the coordination between the domain and range process of the limiting process. The two group members gave similar responses highlighted below in Figure 154.

Table 2

x	f(x)
2.10000000	4.87930340
2.01000000	4.08722195
2.00100000	4.00871339
2.00010000	4.00087125
2.00001000	4.00008712
2.00000100	4.00000871
2.00000010	4.00000087
2.00000001	4.00000009

x	f(x)
1.68000000	6.85900000
1.87800000	7.88059900
1.62000000	7.98800600
1.16200000	7.99880006
1.40500000	7.99988000
1.71600000	7.99998800
1.70100000	7.99999880
1.29000000	7.99999988

Conclusions:

You can't make a conclusion because it is unclear what is happening on [1,2].

The limit as you approach from the left of SOMETHING appears to be 8 and the limit as you approach 2 from the right is 4.

Table 4

x	f(x)
2.10000000	16.11758758
2.01000000	12.50332216
2.00100000	10.07769600
2.00010000	17.57600000
2.00001000	8.21794983
2.00000100	16.71830269
2.00000010	13.16097188
2.00000001	9.48773561

x	f(x)
1.90000000	6.85900000
1.99000000	7.88059900
1.99900000	7.98800600
1.99990000	7.99880006
1.99999000	7.99988000
1.99999900	7.99998800
1.99999990	7.99999880
1.99999999	7.99999988

Conclusions:

The limit as x approaches 2 from the left is 8 but you can't tell what the limit is from the right because of the random list of numbers on the right.

Figure 154. Group N_2's Selected Response from Post-lab 3

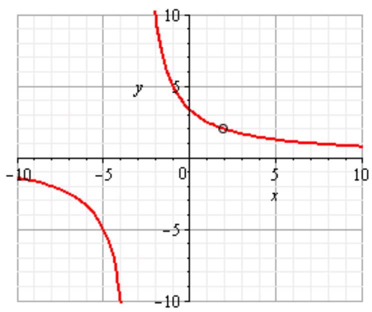
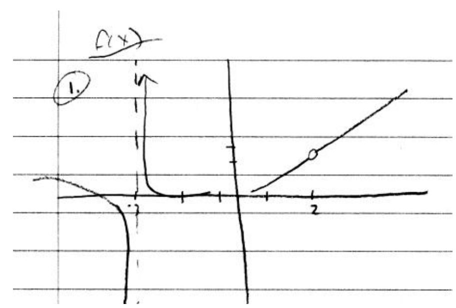
Clearly from these comments, the pair understands there is a lack of coordination between the domain and range processes at play. However, the comment "*The limit as you approach from the left of SOMETHING appears to be 8*" also suggests greater focus on the range process. Their knowledge of this coordination is not fully formed, i.e. it has not been interiorized. Thus the pair appears to have an understanding of APOS steps 3a, 3b, and 3c but continues to refine their understanding of 3c. Moreover, when the instructor asked the students, under what circumstance does a limit exists, the pair gave this telling response with regard to further use of `leftLim` and `rightLim`, "*If the answer continues to decimate itself (ha-ha), then there is a limit. If it doesn't then there isn't a limit."* The pair suggests the limiting process is one in which the limiting value is trapped via an unending process of challenge and response implying a clear sense of coordination.

The pair gave each other perfect peer reviews commenting that they have been friends since middle school and, as a result of their friendship, work together effortlessly. As mentioned in lab two, this pair changes roles like clockwork and does not need to be reminded of the importance of changing roles; they still have a tendency to overlap their respective responsibilities, however. The dynamics of their interaction rarely changes much even after they change roles.

Group N_3

Using the `leftLim` and `rightLim` tools, the pair utilized them to infer the behavior of the four mystery functions from the previous lab.

For function f, see Figure 155, the pair accurately inferred a vertical asymptote at x=-3 as well as a hole at x=2 using the tables produced by `leftLim` and `rightLim`, coupled with evaluation at x=2. The do not discern the decreasing behavior at x=2 however. Their response completely analyzes the functions behavior and demonstrates a clear understanding how the limiting behavior implies these characteristics.

Actual Mystery Function $f(x)$	**Inferred Behavior**
 Hole at x=2, Vertical asymptote at x= -3.	

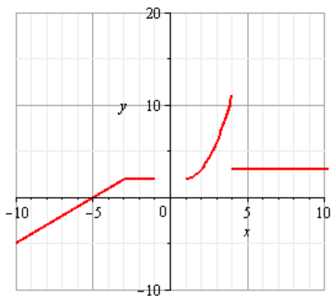
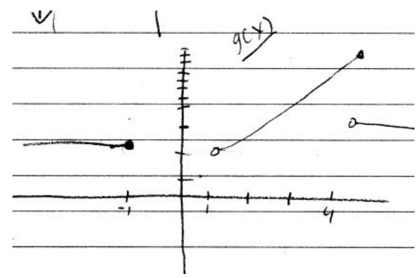
Support at x=-3 and x=2 using `leftLim` and `rightLim`

```
> leftLim(f,-3,5);
[Left Limit]  x              f(x)
--------------- ---------------
     -3.10000000    -100.00000000
     -3.01000000   -1000.00000000
     -3.00100000  -10000.00000000
     -3.00010000  -100000.00000000
     -3.00001000  -1000000.00000000
> rightLim(f,-3,5);
[Right Limit] x              f(x)
--------------- ---------------
     -2.90000000     100.00000000
     -2.99000000    1000.00000000
     -2.99900000   10000.00000000
     -2.99990000  100000.00000000
     -2.99999000  1000000.00000000
```

```
> leftLim(f,2,5);
[Left Limit]  x              f(x)
--------------- ---------------
      1.90000000       2.04081633
      1.99000000       2.00400802
      1.99900000       2.00040008
      1.99990000       2.00004000
      1.99999000       2.00000400
> rightLim(f,2,5);
[Right Limit] x              f(x)
--------------- ---------------
      2.10000000       1.96078431
      2.01000000       1.99600798
      2.00100000       1.99960008
      2.00010000       1.99996000
      2.00001000       1.99999600
> f(2);
      undefined
```

Figure 155. Group N_3's analysis of mystery function f(x) in Lab 3

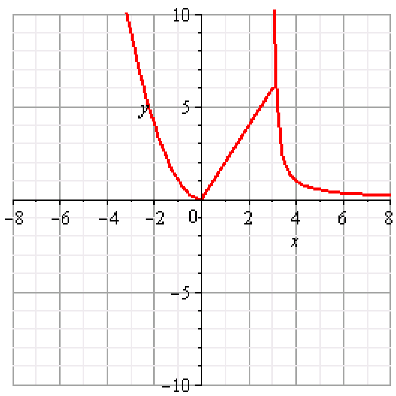
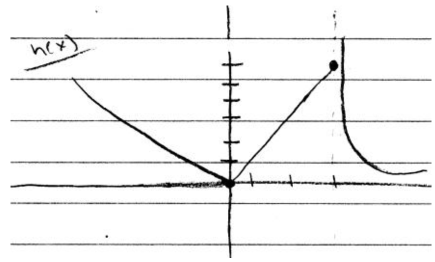
They were similarly successful in their analysis of g, h, and k as shown in Figure 156, Figure 157 and Figure 158. Their graph of g closely resembles the actual graph of g and is clearly supported with output from the tools.

Actual Mystery Function $g(x)$	**Inferred Behavior**
 Jump at x=4, no left limit at x=1, no right limit at x=-1.	
Support at x=-1, 1, and 4 using `leftLim` and `rightLim`	


```
> leftLim(g,-1,4);
[Left Limit]   x              f(x)
---------------- ----------------
    -1.10000000       2.00000000
    -1.01000000       2.00000000
    -1.00100000       2.00000000
    -1.00010000       2.00000000
> g(-1);
     2
> rightLim(g,-1,4);
[Right Limit]  x              f(x)
---------------- ----------------
    -0.90000000              NaN
    -0.99000000              NaN
    -0.99900000              NaN
    -0.99990000              NaN
```

```
> leftLim(g,1,4);
[Left Limit]   x              f(x)
---------------- ----------------
     0.90000000              NaN
     0.99000000              NaN
     0.99900000              NaN
     0.99990000              NaN
> rightLim(g,1,4);
[Right Limit]  x              f(x)
---------------- ----------------
     1.10000000       2.01000000
     1.01000000       2.00010000
     1.00100000       2.00000100
     1.00010000       2.00000001
> g(1);
     undefined
> leftLim(g,4,4);
[Left Limit]   x              f(x)
---------------- ----------------
     3.90000000      10.41000000
     3.99000000      10.94010000
     3.99900000      10.99400100
     3.99990000      10.99940001
> rightLim(g,4,4);
[Right Limit]  x              f(x)
---------------- ----------------
     4.10000000       3.00000000
     4.01000000       3.00000000
     4.00100000       3.00000000
     4.00010000       3.00000000
> g(4);
     11
```

Figure 156. Group N_3's analysis of mystery function g(x) in Lab 3

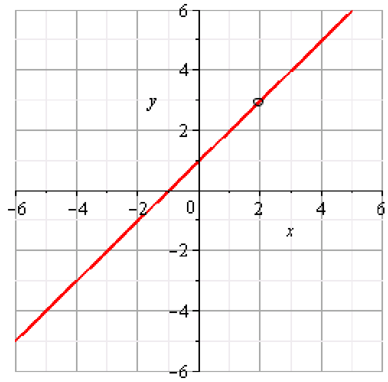
Actual Mystery Function $h(x)$	**Inferred Behavior**
 Corner at x=0 and a Vertical asymptote at x=3.	

Support at x=0 and x=3 using `leftLim` and `rightLim`

```
> leftLim(h,0,4);
[Left Limit]   x              f(x)
---------------- ----------------
    -0.10000000       0.01000000
    -0.01000000       0.00010000
    -0.00100000       0.00000100
    -0.00010000       0.00000001
> rightLim(h,0,4);
[Right Limit] x               f(x)
---------------- ----------------
     0.10000000       0.20000000
     0.01000000       0.02000000
     0.00100000       0.00200000
     0.00010000       0.00020000
```

```
> leftLim(h,3,4);
[Left Limit]   x              f(x)
---------------- ----------------
     2.90000000       5.80000000
     2.99000000       5.98000000
     2.99900000       5.99800000
     2.99990000       5.99980000
> rightLim(h,3,4);
[Right Limit] x               f(x)
---------------- ----------------
     3.10000000      10.00000000
     3.01000000     100.00000000
     3.00100000    1000.00000000
     3.00010000   10000.00000000
> h(0); h(3)
      0
      6
```

Figure 157. Group N_3's analysis of mystery function h(x) in Lab 3

<table>
<tr><th>Actual Mystery Function $k(x)$</th><th>Inferred Behavior</th></tr>
<tr><td>

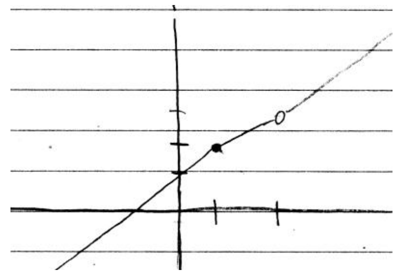
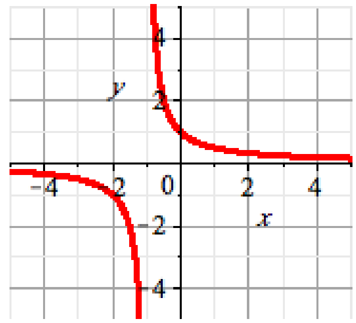
hole at x=2.</td><td></td></tr>
<tr><td colspan="2">Support at x=1 and x=2 using `leftLim` and `rightLim`</td></tr>
<tr><td><pre>
> leftLim(k,1,4);
[Left Limit]   x   f(x)
---------------- ----------------
     0.90000000   1.90000000
     0.99000000   1.99000000
     0.99900000   1.99900000
     0.99990000   1.99990000
> rightLim(k,1,4);
[Right Limit] x   f(x)
---------------- ----------------
     1.10000000   2.10000000
     1.01000000   2.01000000
     1.00100000   2.00100000
     1.00010000   2.00010000
> k(1);
     2
</pre></td><td><pre>
> leftLim(k,2,4);
[Left Limit]   x   f(x)
---------------- ----------------
     1.90000000   2.90000000
     1.99000000   2.99000000
     1.99900000   2.99900000
     1.99990000   2.99990000
> rightLim(k,2,4);
[Right Limit] x   f(x)
---------------- ----------------
     2.10000000   3.10000000
     2.01000000   3.01000000
     2.00100000   3.00100000
     2.00010000   3.00010000
> k(2);
     undefined
</pre></td></tr>
</table>

Figure 158. Group N_3's analysis of mystery function k(x) in Lab 3

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple that have specified behavior. As shown in Figure 159, the pair accomplished the task of creating a function possessing a hole at x=2 and a vertical asymptote at x=-1. They were successful in creating a suitable function in Maple and provided clear justification using the `leftLim` and `rightLim` tools.

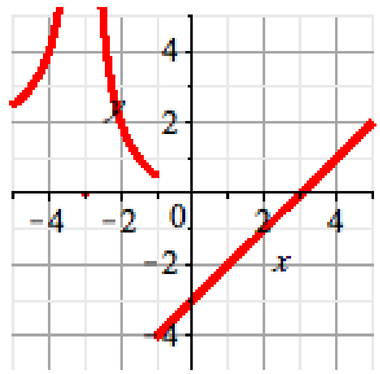
Desired Behavior	**Function definition**	**Plot**
Hole at x= 2 Vertical Asymptote at x=-1	`b:=x->((x-2)/(x^2-x-2));`	

Supporting data produced with tools.	`> leftLim(b,-1,4);`
	`[Left Limit]   x   f(x)`
	`-----------  ------------`
	`-1.100000    -10.000000`
	`-1.010000   -100.000000`
	`-1.001000  -1000.000000`
	`-1.000100 -10000.000000`
	`> rightLim(b,-1,4);`
	`[Right Limit] x   f(x)`
	`-----------  ------------`
	`-0.900000     10.000000`
	`-0.990000    100.000000`
	`-0.999000   1000.000000`
	`-0.999900  10000.000000`
		`> b(2);`
		`Error, (in b) numeric exception: division by zero`
		`> leftLim(b,2,4);`
		`[Left Limit]   x   f(x)`
		`------------ ----------`
		`1.900000      0.344827`
		`1.990000      0.334448`
		`1.999000      0.333444`
		`1.999900      0.333344`
		`> rightLim(b,2,4);`
		`[Right Limit] x   f(x)`
		`------------ ----------`
		`2.100000      0.322580`
		`2.010000      0.332225`
		`2.001000      0.333222`
		`2.000100      0.333322`

Figure 159. Group N_3's construction of function b(x) in Lab 3

Again, for function c, the pair produced all of the requested behavior. They effectively constructed an appropriate procedure and provided support for the functions behavior at x=-1 and x=-3. The results are shown in Figure 160.

Desired Behavior	Function definition	Plot

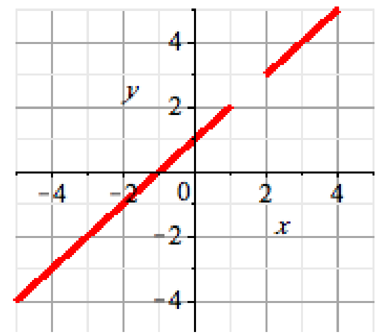
Jump discontinuity at x= -1
$\lim_{x\to -3^-} c(x) = +\infty$
$\lim_{x\to -3^+} c(x) = +\infty$
	`c:= x->piecewise(`
	` x<-3, x/(x+3),`
	` x>-3 and x<-1, -x/(x+3),`
	` x>=-1, x-3);`
		

Supporting data produced with tools.	`> leftLim(c,-1,4);`
	`[Left Limit]  x    f(x)`
	`--------------- ---------`
	`-1.100000      0.578947`
	`-1.010000      0.507537`
	`-1.001000      0.500750`
	`-1.000100      0.500075`
	`> rightLim(c,-1,4);`
	`[Right Limit] x    f(x)`
	`--------------- ---------`
	`-0.900000     -3.900000`
	`-0.990000     -3.990000`
	`-0.999000     -3.999000`
	`-0.999900     -3.999900`
		`> leftLim(c,-3,4);`
		`[Left Limit]  x    f(x)`
		`--------------- -----------`
		`-3.100000      31.000000`
		`-3.010000     301.000000`
		`-3.001000    3001.000000`
		`-3.000100   30001.000000`
		`> rightLim(c,-3,4);`
		`[Right Limit] x   f(x)`
		`--------------- -----------`
		`-2.900000      29.000000`
		`-2.990000     299.000000`
		`-2.999000    2999.000000`
		`-2.999900   29999.000000`

Figure 160. Group N_3's construction of function c(x) in Lab 3

For function d, the pair was able to construct the requested behaviors. As with the previous functions, the pair provided appropriate justification using `leftLim` and `rightLim` tools (see Figure 161). They did inquire about the error message reported in response to the left hand limit at x=2 and right hand limit at x=1. The instructor explained why this occurred and indicated that it was an oversight in the development of the tool.

Desired Behavior	**Function definition**	**Plot**

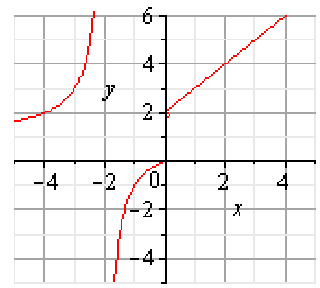
Undefined on [1,2]
$\lim_{x\to1^-} d(x) = 2$
$\lim_{x\to2^+} d(x) = 3$
	`d:= x-> piecewise(`
	`x<1, x+1,`
	`x>=1 and x<=2,`
	`undefined,`
	`x>2, x+1);`
		

Supporting data produced with tools.	`> rightLim(d,1,4);`
	`[Right Limit] x`
	`f(x)`
	`---------- ----------`
	`1.10000000`
	
	`Error, (in fprintf)`
	`number expected for`
	`floating point format`
	`> leftLim(d,2,4);`
	`[Left Limit]  x f(x)`
	`---------- ----------`
	`1.90000000`
	
	`Error, (in fprintf)`
	`number expected for`
	`floating point format`
		`> leftLim(d,1,4);`
		`[Left Limit]  x   f(x)`
		`------------- ------------`
		`-`
		`0.90000000      1.90000000`
		`0.99000000      1.99000000`
		`0.99900000      1.99900000`
		`0.99990000      1.99990000`
		`> rightLim(d,2,4);`
		`[Right Limit] x    f(x)`
		`------------- ------------`
		`-`
		`2.10000000      3.10000000`
		`2.01000000      3.01000000`
		`2.00100000      3.00100000`
		`2.00010000      3.00010000`

Figure 161. Group N_3's construction of function d(x) in Lab 3

Function e satisfied all the criteria specified and was supported with the appropriate application of `leftLim`, `rightLim`, and evaluation (see Figure 162).

Desired Behavior	Function definition	Plot

$\lim_{x \to 1} e(x) = 3$
$e(1) = 5$
$\lim_{x \to -2^-} e(x) = +\infty$
$\lim_{x \to -2^+} e(x) = -\infty$
	`e:= x-> piecewise(`
	`x>-5 and x<0,    x/(x+2),`
	`x>=0 and x<1, x+2,`
	`x>1, x+2,`
	`x=1,5,`
	`undefined);`
		

Supporting data produced with tools.	`> leftLim(e,-2,5);`
	`[Left Limit]   x    f(x)`
	`------------- -------------`
	`-`
	`-2.100000     21.000000`
	`-2.010000    201.000000`
	`-2.001000   2001.000000`
	`-2.000100  20001.000000`
	`-2.000010 200001.000000`
	`> rightLim(e,-2,5);`
	`[Right Limit] x     f(x)`
	`------------ --------------`
	`-`
	`-1.900000    -19.000000`
	`-1.990000   -199.000000`
	`-1.999000  -1999.000000`
	`-1.999900 -19999.000000`
	`-1.999990 -199999.000000`
	`> e(1);`
	`5`
		`> leftLim(e,1,4);`
		`[Left Limit]   x   f(x)`
		`--------     ----------`
		`0.900000      2.900000`
		`0.990000      2.990000`
		`0.999000      2.999000`
		`0.999900      2.999900`
		`> rightLim(e,1,4);`
		`[Right Limit] x   f(x)`
		`-----------   ---------`
		`1.100000      3.100000`
		`1.010000      3.010000`
		`1.001000      3.001000`
		`1.000100      3.000100`

Figure 162. Group N_3's construction of function e(x) in Lab 3

Notably, as with group N_2, the pair utilized the `leftLim` and `rightLim` tools for both *analysis and synthesis* of the requested functions. When asked to create functions with stated limiting behavior, the pair was observed utilizing the tool to aid in the construction of the functions on paper. The pair experimented with candidate functions using the tool to test their conjectures and incrementally build their resulting functions. The pair gave each other perfect peer reviews and commented that this was their favorite lab thus far.

Following the lab, the group completed a written post-lab activity to explore the degree to which they understood the coordination between the domain and range process of the limiting process. The two group members gave similar responses shown in Figure 163.

Clearly, from these comments, the pair quite clearly understands there exists a lack of coordination between the domain and range processes; In fact, prior to analysis, the pair reordered the table in increasing order of x! Thus, the pair possesses a clear understanding of APOS steps 3a, 3b, and 3c.

Post-Lab 3: Relevant Student Responses

Table 2		**Table 8**	
x	f(x)	x	f(x)
2.10000000	4.87930340	3.03300000	29.79100000
2.01000000	4.08722195	3.03500000	27.00000270
2.00100000	4.00871339	3.39700000	27.27090100
2.00010000	4.00087125	3.41400000	27.00270009
2.00001000	4.00008712	3.41900000	27.00000027
2.00000100	4.00000871	3.54600000	27.00027000
2.00000010	4.00000087	3.58900000	27.02700900
2.00000001	4.00000009	3.62800000	27.00002700
x	f(x)	x	f(x)
1.68000000	6.85900000	2.08900000	26.99997300
1.87800000	7.88059900	2.41800000	26.99999730
1.62000000	7.98800600	2.44800000	26.73089900
1.16200000	7.99880006	2.58100000	26.99999973
1.40500000	7.99988000	2.62000000	26.97300900
1.71600000	7.99998800	2.75900000	24.38900000
1.70100000	7.99999880	2.87000000	26.99730009
1.29000000	7.99999988	2.93300000	26.99973000

Table 2 — Conclusions:
The first table suggests $\lim_{x \to 2^+} f(x) = 4$. For the second table, *no conclusion b/c you can't tell what happens between (1,2].*

Table 8 — Conclusions:

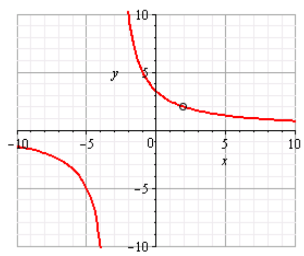
The first table does not accurately show what the limit is, and the second table is the same way.

Figure 163. Group N_3's Selected Response from Post-lab 3

Group N_4

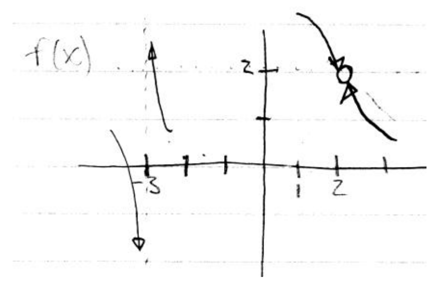
Using the `leftLim` and `rightLim` tools shown in Figure 164, the pair inferred the behavior of the four mystery functions from the previous lab. The pair accurately inferred a vertical asymptote at x=-3 using the resulting tables shown. Additionally, they further inferred the presence of a hole at x=2 using `leftLim`, `rightLim`, and evaluation at x=2. This response demonstrates effective tool use and clear understanding of the limit process.

Actual Mystery Function $f(x)$



Hole at x=2, Vertical asymptote at x= -3.

Inferred Behavior



Support at x=-3 and x=2 using `leftLim` and `rightLim`

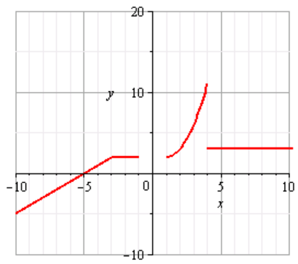
```
> leftLim(f, -3, 4);
[Left Limit]   x              f(x)
---------------- ----------------
     -3.10000000    -100.00000000
     -3.01000000   -1000.00000000
     -3.00100000  -10000.00000000
     -3.00010000 -100000.00000000
> rightLim(f, -3, 4);
[Right Limit]  x              f(x)
---------------- ----------------
     -2.90000000     100.00000000
     -2.99000000    1000.00000000
     -2.99900000   10000.00000000
     -2.99990000  100000.00000000
```

```
> leftLim(f, 2, 4);
[Left Limit]   x              f(x)
---------------- ----------------
      1.90000000       2.04081633
      1.99000000       2.00400802
      1.99900000       2.00040008
      1.99990000       2.00004000
> rightLim(f, 2, 4);
[Right Limit]  x              f(x)
---------------- ----------------
      2.10000000       1.96078431
      2.01000000       1.99600798
      2.00100000       1.99960008
      2.00010000       1.99996000
> f(2);
      undefined
```

Figure 164. Group N_4 's analysis of mystery function f(x) in Lab 3

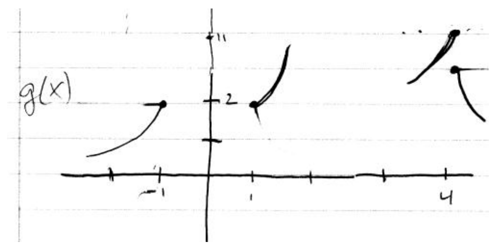
They were similarly successful in their analysis of g, h, and k as shown. Their graph of g closely resembles the actual graph of g in terms of limiting behavior.

Actual Mystery Function $g(x)$



Jump at x=4, no left limit at x=1, no right limit at x=-1.

Inferred Behavior



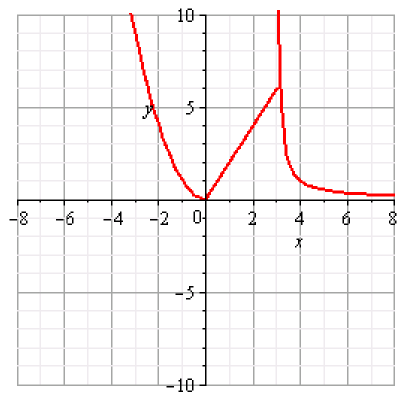
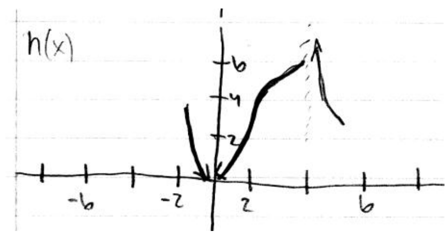
Support at x=-1, 1, and 4 using `leftLim` and `rightLim`

```
> leftLim(g, -1, 4);
[Left Limit]   x              f(x)
--------------- ---------------
    -1.10000000      2.00000000
    -1.01000000      2.00000000
    -1.00100000      2.00000000
    -1.00010000      2.00000000
> rightLim(g, -1, 4);
[Right Limit]  x              f(x)
--------------- ---------------
    -0.90000000             NaN
    -0.99000000             NaN
    -0.99900000             NaN
    -0.99990000             NaN
> leftLim(g, 1, 4);
[Left Limit]   x              f(x)
--------------- ---------------
     0.90000000             NaN
     0.99000000             NaN
     0.99900000             NaN
     0.99990000             NaN
> rightLim(g, 1, 4);
[Right Limit]  x              f(x)
--------------- ---------------
     1.10000000      2.01000000
     1.01000000      2.00010000
     1.00100000      2.00000100
     1.00010000      2.00000001
```

```
> leftLim(g, 4, 4);
[Left Limit]   x              f(x)
--------------- ---------------
     3.90000000     10.41000000
     3.99000000     10.94010000
     3.99900000     10.99400100
     3.99990000     10.99940001
> rightLim(g, 4, 4); g(5);
[Right Limit]  x              f(x)
--------------- ---------------
     4.10000000      3.00000000
     4.01000000      3.00000000
     4.00100000      3.00000000
     4.00010000      3.00000000

     3
```

Figure 165. Group N_4's analysis of mystery function g(x) in Lab 3

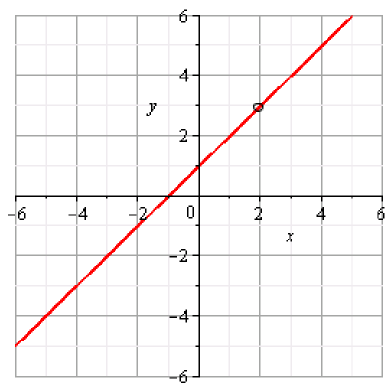
The pair again makes accurate descriptions of functions h(x) (see Figure 166) and k(x)’s (see Figure 167) behavior using the resulting limit tables and evaluations.

Actual Mystery Function $h(x)$	**Inferred Behavior**
	
Corner at x=0 and a Vertical asymptote at x=3.	
Support at x=0 and x=3 using `leftLim` and `rightLim`	

`> leftLim(h, 0, 4);`
`[Left Limit]   x              f(x)`
`---------------  ---------------`
`    -0.10000000       0.01000000`
`    -0.01000000       0.00010000`
`    -0.00100000       0.00000100`
`    -0.00010000       0.00000001`
`> rightLim(h, 0, 4);`
`[Right Limit] x              f(x)`
`---------------  ---------------`
`     0.10000000       0.20000000`
`     0.01000000       0.02000000`
`     0.00100000       0.00200000`
`     0.00010000       0.00020000`
	`> leftLim(h, 3, 4);`
	`[Left Limit]   x              f(x)`
	`---------------  ---------------`
	`     2.90000000       5.80000000`
	`     2.99000000       5.98000000`
	`     2.99900000       5.99800000`
	`     2.99990000       5.99980000`
	`> rightLim(h, 3, 4);`
	`[Right Limit] x              f(x)`
	`---------------  ---------------`
	`     3.10000000      10.00000000`
	`     3.01000000     100.00000000`
	`     3.00100000    1000.00000000`
	`     3.00010000   10000.00000000`

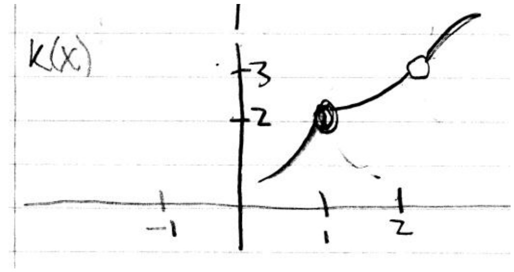
Figure 166. Group N_4 's analysis of mystery function h(x) in Lab 3

Actual Mystery Function $k(x)$



hole at x=2.

Inferred Behavior



Support at x=1 and x=2 using `leftLim` and `rightLim`

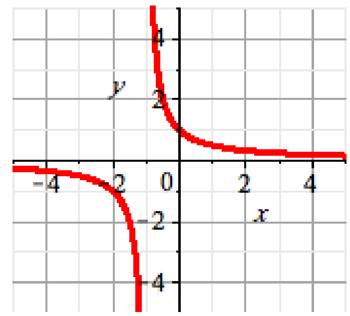
```
> leftLim(k, 1, 4);
[Left Limit]   x               f(x)
---------------  ----------------
     0.90000000       1.90000000
     0.99000000       1.99000000
     0.99900000       1.99900000
     0.99990000       1.99990000
> rightLim(k, 1, 4);
[Right Limit]  x               f(x)
---------------  ----------------
     1.10000000       2.10000000
     1.01000000       2.01000000
     1.00100000       2.00100000
     1.00010000       2.00010000
```

```
> leftLim(k, 2, 4);
[Left Limit]   x               f(x)
---------------  ----------------
     1.90000000       2.90000000
     1.99000000       2.99000000
     1.99900000       2.99900000
     1.99990000       2.99990000
> rightLim(k, 2, 4);
[Right Limit]  x               f(x)
---------------  ----------------
     2.10000000       3.10000000
     2.01000000       3.01000000
     2.00100000       3.00100000
     2.00010000       3.00010000
```

Figure 167. Group N_4's analysis of mystery function k(x) in Lab 3

The graph sketched closely reflects the limiting behavior of the function. One notable omission, however, is that the group did not evaluate function k at x=1 nor x=2 and is not justified in including the point at x=1 and drawing a hole at x=2.

In the second half of the lab, rather than explore a given functions behavior, pairs were instructed to construct functions in Maple that have specified behavior. As shown in Figure 168, the pair accomplished the task of creating a function possessing a hole at x=2 and a vertical asymptote at x=-1. They were successful in creating a suitable function and provided clear justification of the hole but not the vertical asymptote using the `leftLim` and `rightLim` tools.

Desired Behavior	**Function definition**	**Plot**
Hole at x= 2 Vertical Asymptote at x=-1	`b:=x->((x-2)/(x^2-x-2));`	

Supporting data produced with tools.	`> leftLim(b, 2, 5);`
	`[Left Limit]  x      f(x)`
	`------------  ------------`
	`1.90000000      .34482759`
	`1.99000000      .33444816`
	`1.99900000      .33344448`
	`1.99990000      .33334444`
	`1.99999000      .33333444`
	`> rightLim(b, 2, 5);`
	`[Right Limit] x      f(x)`
	`------------- -------------`
	`2.10000000      .32258065`
	`2.01000000      .33222591`
	`2.00100000      .33322226`
	`2.00010000      .33332222`
	`2.00001000      .33333222`
		`> b(2);`
		`Error, (in b) numeric exception: division by zero`

Figure 168. Group N_4's construction of function b(x) in Lab 3

For function c (see Figure 169), the pair correctly produced and defined a function with the requested jump discontinuity at x=-1 without justification using `leftLim` or `rightLim`. Further, they were not successful in creating asymptotic behavior at x=-3. Although the graph looks as if it possessed an asymptote x=-3, the pair utilized a pair of cubic polynomials to create this behavior. Closer exploration would have revealed these cubic graphs eventually intersect.

As shown, the pair attempted to justify the requested asymptotic behavior at x=-3, but the tables do not suggest nor support this behavior; the function values do not appear to become arbitrarily large.

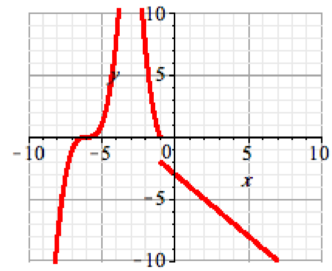
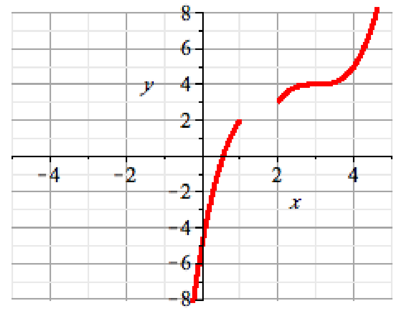
<table>
<tr><th>Desired Behavior</th><th>Function definition</th><th>Plot</th></tr>
<tr><td>Jump discontinuity at x= -1
$\lim_{x\to -3^-} c(x) = +\infty$
$\lim_{x\to -3^+} c(x) = +\infty$</td><td>c:=x -> piecewise(
x<-3, (x+6)^3,
x>-3 and x<-1, -x^3-1,
x>=-1, -x-3,
undefined);</td><td>
</td></tr>
<tr><td>Supporting data produced with tools.</td><td colspan="2">> leftLim(c, -3, 4);
[Left Limit] x f(x)
--------------- -----------
-3.10000000 24.38900000
-3.01000000 26.73089900
-3.00100000 26.97300900
-3.00010000 26.99730009
> rightLim(c,-3, 4);
[Right Limit] x f(x)
--------------- -----------
-2.90000000 23.38900000
-2.99000000 25.73089900
-2.99900000 25.97300900
-2.99990000 25.99730009</td></tr>
</table>

Figure 169. Group N_4's construction of function c(x) in Lab 3.

For function d, see Figure 170, the pair was able to accomplish the requested domain restriction as well as construct the requested limiting behavior. The pair provided appropriate justification using `leftLim` and `rightLim` tools for all behavior except the domain restriction.

Desired Behavior	**Function definition**	**Plot**

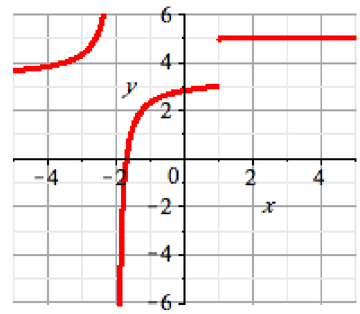
Undefined on [1,2]
$\lim_{x\to1^-} d(x)=2$
$\lim_{x\to2^+} d(x)=3$
	`d := x -> piecewise(`
	`x<1, (x-2)^3+3,`
	`x>2, (x-3)^3+4,`
	`undefined);`
		

Supporting data produced with tools.	`> leftLim(d, 1, 4;`
	`[Left Limit]   x   f(x)`
	`---------------- ------`
	`0.900000      1.669000`
	`0.990000      1.969699`
	`0.999000      1.996997`
	`0.999900      1.999699`
	`> rightLim(d,2,4;`
	`[Right Limit] x   f(x)`
	`---------------- ------`
	`2.100000      3.271000`
	`2.010000      3.029701`
	`2.001000      3.002997`
	`2.000100      3.000299`
		`> d(1);`
		`undefined`
		`> d(2);`
		`undefined`

Figure 170. Group N_4's construction of function d(x) in Lab 3

Function e (see Figure 171) satisfied all the criteria except the limiting behavior at x=1 but was not fully supported with the application of `leftLim` and `rightLim` and evaluation.

Desired Behavior	Function definition	Plot

$\lim_{x\to 1} e(x) = 3$
$e(1) = 5$
$\lim_{x\to -2^-} e(x) = +\infty$
$\lim_{x\to -2^+} e(x) = -\infty$
	`e := x -> piecewise(`
	`x<1, (-1/(x+2))+3+(1/3),`
	`x>=1,5);`
		

Supporting data produced with tools.	`> e(1);`
	`5`
	`> rightLim(e,-2,4);`
	`[Right Limit] x f(x)`
	`--------------- ----------`
	`-1.90000000 -6.66666667`
	`-1.99000000 -96.66666667`
	`-1.99900000 -996.66666667`
	`-1.99990000 -9996.66666667`
		`> leftLim(e,1,4);`
		`[Left Limit] x f(x)`
		`---------------- ----------`
		`0.90000000 2.98850575`
		`0.99000000 2.99888517`
		`0.99900000 2.99988885`
		`0.99990000 2.99998889`

Figure 171. Group N_4's construction of function e(x) in Lab 3

Following the lab, see Figure 172, the group completed a written post-lab activity to explore the degree to which they understood the coordination between the domain and range process of the limiting process. The two group members gave similar responses.

Clearly from their comments, the pair very clearly understands there is must be a coordination between the domain and range processes The pair possess a clear understanding of APOS steps 3a, 3b, and 3c.

One group member gave a perfect peer evaluation of the other. The other indicated their partner did not always cooperatively follow the pair-programming model giving them sub-score of 19/20; the instructor had to prod the group to change roles on several occasions as one student tended to monopolize time at the computer.

<table>
<tr><th colspan="2">Post-Lab 3: Relevant Student Responses</th></tr>
<tr><td>

Table 2

x	f(x)
2.10000000	4.87930340
2.01000000	4.08722195
2.00100000	4.00871339
2.00010000	4.00087125
2.00001000	4.00008712
2.00000100	4.00000871
2.00000010	4.00000087
2.00000001	4.00000009

x	f(x)
1.68000000	6.85900000
1.87800000	7.88059900
1.62000000	7.98800600
1.16200000	7.99880006
1.40500000	7.99988000
1.71600000	7.99998800
1.70100000	7.99999880
1.29000000	7.99999988

Conclusions:

You can tell that $\lim_{x\to 2^+} f(x) = 4$ *by looking at the first limit table, but you can't tell what is going on in the second table b/c the x-values jump all over the place.*

</td><td>

Table 4

x	f(x)
2.10000000	16.11758758
2.01000000	12.50332216
2.00100000	10.07769600
2.00010000	17.57600000
2.00001000	8.21794983
2.00000100	16.71830269
2.00000010	13.16097188
2.00000001	9.48773561

x	f(x)
1.90000000	6.85900000
1.99000000	7.88059900
1.99900000	7.98800600
1.99990000	7.99880006
1.99999000	7.99988000
1.99999900	7.99998800
1.99999990	7.99999880
1.99999999	7.99999988

Conclusions:

As x approaches 2 from the left the y-value nears 8 but when the x-value approaches 2 from the right, the y-value jumps all over the place. i.e.
$\lim_{x\to 2^-} f(x) = 8$ *and* $\lim_{x\to 2^+} f(x) = DNE$

</td></tr>
</table>

Figure 172. Group N_4's post-lab responses to Lab 3

Tool Use and Justification in Lab 3

To assess the extent to which the `leftLim` and `rightLim` tools were being utilized by the groups, a tally of the number of times the tool was used successfully to justify limiting behavior in lab three was compiled. Groups were asked to construct four functions, $b(x), c(x), d(x),$ and $e(x)$, with a total of 12 specific characteristics. The bar chart shown in Figure 173 indicates a tally of the number of characteristics created, the upper bar, and the number of characteristics justified using the tool, the lower bar.

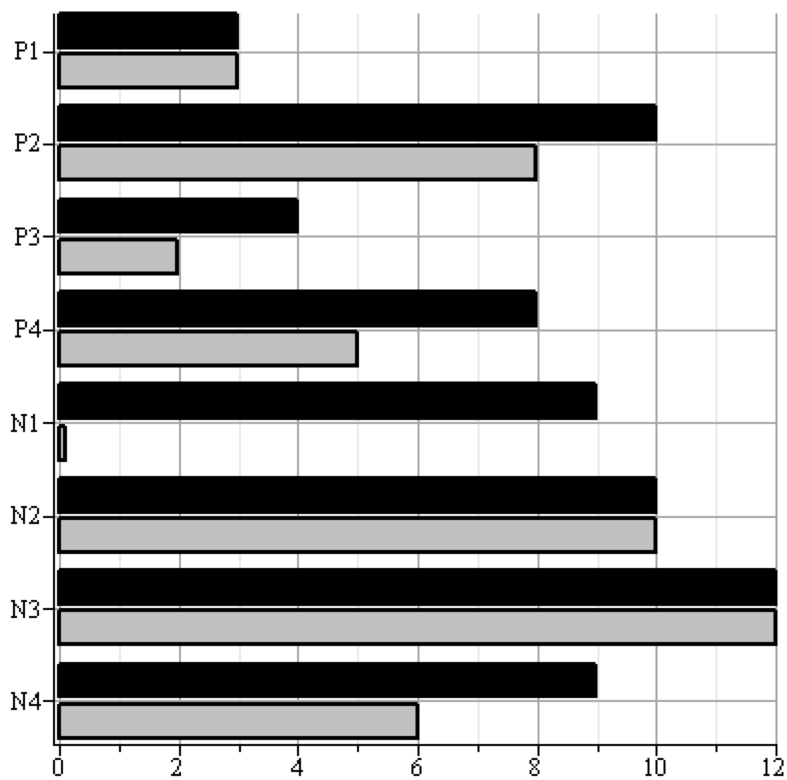


Figure 173. Tool Usage in Lab 3. The upper bar indicates the number of characteristics successfully created. The lower bar indicates the number of successful justifications using the `leftLim` and `rightLim` tools.

Clearly the non-programming group N was more successful than the programming group P in terms of the number of successfully created behaviors and in terms of the number of justifications successfully accomplished. Although programming group P_2 was also successful. The most successful groups in the study are those that have adopted and utilized the tools.

Lab 3 Summary

This lab focused upon the coordination of the domain and range processes, APOS Step 3c, of the APOS decomposition through the construction and use of two tools, `leftLim` and `rightLim`. Groups were given the opportunity to create a suitable domain process, APOS Step 3a, and an understanding of the resulting range process, APOS Step 3b, through the design and construction of functions with prescribed limiting behavior. Groups were required to support these constructed behaviors using the `leftLim` and `rightLim` tools.

One sided limits where specifically considered so as to deepen the understanding of the domain processes and the need for coordination with the resulting range process. The degree to which pairs understood the coordination between the domain and range process of the limiting process, APOS step 3c, was subsequently assessed in the postlab activity.

Additionally, the non-programming groups, class N, developed an understanding of how to define piecewise functions in Maple and the programming groups, class P, learned to implement piecewise functions using procedures with conditional behavior and the for-loop looping construct.

Programming groups, $P_1, P_3,$ and P_4, failed to constructively utilize the `leftLim`

and `rightLim` procedures although all groups were successful in their implementation. Group P_3 appeared to be able to interpret the output from the tools but either did not apply it to the correct function or was unable to implement the desired limiting behaviors due to programming difficulties and misunderstandings. These three groups all mistakenly defined a squaring function and attempted to analyze it rather than the intended mystery functions. Assumedly this was caused by confusion with prelab activities; prior to the lab, the implementation of the squaring function was used as an example of procedural function definition with the P groups. These groups failed to understand that they were to analyze the mystery functions that I could not entice them to explore in Lab 2.

Beyond the confusion relating to which functions were to be analyzed, even when participants could utilize the tool, the P groups rarely made correct inferences using the resulting output. Groups P_2 and P_3 were the only programming groups who were able to correctly interpret the output from the tools. Group P_2 was the only programming group to effectively utilize the tools by applying them to the intended functions.

It appeared the P groups were cognitively overloaded by the combination of having to implement and utilize the two tools. This confusion obscured the intended goals of the lab leading participants to see the creation of the tool as the end rather than the analysis of the requested functions. Groups P_2 and P_3 were the only programming groups to achieve APOS Step 3c on this lab. They were the only pairs to develop an ability to interpret the output from the `leftLim` and `rightLim` tools thus understanding the coordination between the domain and range processes.

The non-programming groups were much more successful in the analysis and

application of the `leftLim` and `rightLim` tools. These students had a much clearer idea of what was being requested in the lab as well as how this information was to be discerned. Group N_1 did not understand how to use the tool and, as a result, could not produce relevant data from which to draw conclusions. Groups N_2 and N_3 were both very successful on this lab being able to analyze, create, and justify the behavior of the provided functions using the tools. Group N_4 was more successful with the analysis of the mystery functions than with the construction of functions having the requested properties. Of particular interest was the observation that both groups N_2 and N_3 utilized the tools to help synthesize functions b,c,d, and e. These were the only two pairs in either class N or P to utilize the tool in this manner.

The progression towards a complete understanding of limit within the APOS decomposition is shown for all group pairs in Figure 174. The vertical line highlights the intended level of attainment within the decomposition at the completion of the lab and the shading indicates the group's actual level of attainment.

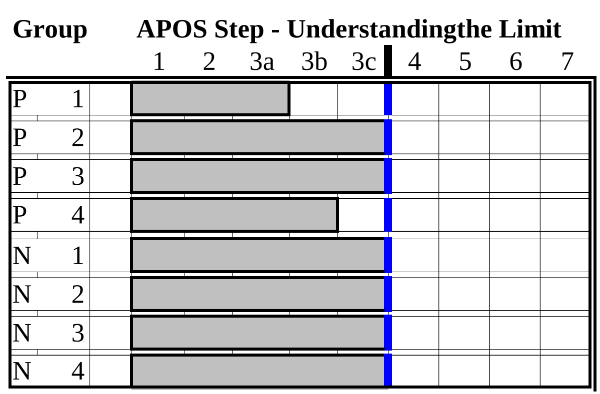


Figure 174. APOS Level of Attainment following Lab 3. The shaded region represents actual attainment and the vertical line indicates the intended level of attainment.

Conceptions of Limit Following Lab 3

At the completion of this lab, there was a noticeable change in the conceptions of limit held by the two classes P and N. The two most frequent definitions selected being the dynamic-theoretical definition and the unreachable definition. At this juncture, more of the non-programming pairs, class N, have adopted the dynamic-theoretic conception while a majority of the programming pairs, class P, have abandoned the dynamic theoretic definition in favor of the unreachable conception.

Group		**Limit Conception**
P_1	H	1
	H	2
P_2	H	4
	L	4
P_3	H	4
P_4	H	3
N_1	L	1
	L	1
N_2	H	1
	L	1
N_3	H	1
	L	1
N_4	L	6
	L	1

Conceptions of Limit (Williams, 1991)

1) (*Dynamic-theoretical*) A limit describes how a function moves as x moves towards a certain point.
2) (*Acting as a boundary*) A limit is a number or point past which a function cannot go.
3) (*Formal*) A limit is a number that the y-values of a function can be made arbitrarily close to by restricting x-values.
4) (*Unreachable*) A limit is a number or point the function gets close to but never reaches.
5) (*Acting as an approximation*) A limit is an approximation that can be made as accurate as you wish.
6) (*Dynamic-practical*) A limit is determined by plugging in numbers closer and closer to a given number until the limit is reached.

H = Higher-performing, L = Lower-performing, X = no response

Figure 175. Postlab 3 Conceptions of Limit

Lab 4 Results

The fourth Maple lab took place during the thirteen-seventeenth weeks, and was the culminating lab experience in which participants were asked to gain an understanding of the formal mathematical definition of limit.

This lab involved a tool, `dePlot`, shown in Figure 176, which created a visual depiction of the inequalities in the formal definition. Using this tool, students visually interacted with the formal definition by interactively determining suitable choices for e (ε in the formal definition), and d (δ in the formal definition). The lab specifically focused on developing an understanding of the coordinated domain and range processes as interval processes, Steps 5 and 6 of the APOS decomposition. By considering ways students applied the tool, an understanding of their understanding of the attendant interval process and their mutual coordination's could be discerned.

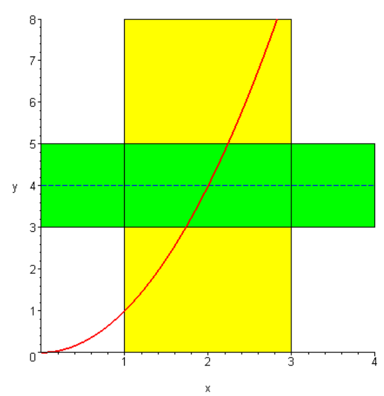
The dePlot tool	**Formal Definition of Limit**
> **dePlot(f, L, ε, c, δ);**  $f(x)=x^2$, L=4, $\varepsilon=1$, **c=2**, $\delta=1$	Let c and L be real numbers. The function f **has limit** L **as** x **approaches** c if, given any positive number ε, there is a positive number δ such that for all x, $$0<\lvert x-c\rvert<\delta \;\Rightarrow\; \lvert f(x)-L\rvert<\varepsilon$$ We write $$\lim_{x\to c} f(x)=L$$
ESSENTIAL IDEA: The inequality $0<\lvert x-c\rvert<\delta$ is satisfied for all x values in the tall rectangle and the inequality $\lvert f(x)-L\rvert<\varepsilon$ is satisfied whenever y is within the horizontal rectangle. Thus students must come to understand that a limit as x->c is determined by trapping the graph within these two rectangles. If this is always possible no matter how narrow the horizontal rectangle is made, then L must be the limit.	

Figure 176. The `dePlot` tool for exploring the formal $\varepsilon-\delta$ definition of limit.

Pairs were asked to provide convincing arguments using the `dePlot` tool that the limit statements $\lim_{x\to 4}\sqrt{x}=2$, $\lim_{x\to 4}\sqrt{x}\neq 1.99$, and $\lim_{x\to 2}x^2\neq 3$ were true. They were also asked to determine whether the limit $\lim_{x\to 1}\frac{|x-1|}{x-1}$ existed or failed to exist. The goal of these questions was to explore student understanding of the coordination of domain and range processes. Students were asked to determine the largest d tolerance required to ensure a stated e tolerance for the limit statements $\lim_{x\to 4}2x-1=7$ for (e=0.5), $\lim_{x\to 4}\frac{x^2-16}{x-4}=8$ for (e=0.5 and e=0.05), and $\lim_{x\to 0}x\sin\left(\frac{1}{x}\right)$ for (e=0.05). The last limit did not have a suggested limiting value and students had to first determine that limiting value.

After completing the lab, a written post-lab activity was completed. Copies of these activities appear in the appendix. Groups responded to three major questions involving interpreting plots produced by `dePlot` exploring student understanding of the tool and the way in which its use reflects the specific mathematical notions of the formal definition of limit.

On the first question, students were given four plots to consider. In each of these, they were asked to interpret `dePlots` in terms of limits. First, a plot that clearly demonstrated that the value 7 was not the limit of function f(x) as x approached 2 was provided. Next, students were given two `dePlots` of the same function that suggested a particular limiting value. They were asked to indicate in mathematical terms what these two plots were suggesting. Specifically, could students translate the plots into corresponding limit statements? The third plot showed function g(x) which appeared inside both shaded bands. The question was whether this proved a particular limiting

value. Finally, a `dePlot` showed a particular e tolerance was not attainable as x approached a particular value. Students were asked to explain this in terms of the functions behavior within the shaded bands in `dePlot`s.

The second major topic explored was whether students understood the connection between `dePlots` and the limit tables explored in Lab 3. Specifically, how these relate to the domain and range processes involved in establishing limits. Here a pair of one-sided limit tables was given along with two corresponding `dePlots`. Students were asked what, if anything, the *columns* of the limit plots had to do with the *shaded rectangles* in the `dePlots`?

The final question presented the students with the correct formal definition of the limit. Students where to explain how the shaded rectangles in a given `dePlot` related to specific phrases in the formal definition.

Group P_1

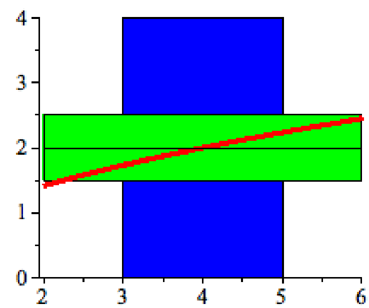
This group was able to develop the tool as requested and demonstrate its use. However, there was a minor error in the drawing of the vertical d-band highlighted in their implementation of `dePlot` in Figure 177 which eventually failed to provide relevant output.

```
dePlot := proc( f, L, e, a, d)
   # declare any necessary local variables here
   #
   local p1,r1,l1,r2;
   # include graphics tools (display, and rectangle)
   #
   with( plottools ):  with( plots ):

   # plot the graph
   p1 := plot( f(x), x=a-2*d..a+2*d,y=L-4*e..L+4*e, discont=true,
thickness=2);
   r1:= rectangle ([a-2*d,L+e],[a+2*d, L-e],color=green);
   l1 :=line( [a-2*d, L], [a+2*d, L], color=black );
   r2:= rectangle ([a-d, 4*e+L],[a+d, 4*e-L],color=blue);
   # display the graph
   display([p1,r1,l1,r2],view=[a-2*d..a+2*d,L-4*e..L+4*e]);
end proc:
```

Figure 177. Group P_1's implementation of the `dePlot` tool.

The group was, however, unable to utilize the tool to make limiting arguments. Specifically, on question 2, the students were asked to use the tool to demonstrate the plausibility of $\lim_{x\to 4}\sqrt{x}=2$. The pair correctly defined the function and properly called the tool (see Figure 178). However, the pair did not produce a sequence of plots suggesting the validity of the claim. They merely produced a single plot demonstrating it is possible to achieve a tolerance of e=0.5 units from 2 by restricting x to within d=1 unit of 4 (see Figure 178). Moreover, their method of argumentation using this tool as well as their interpretation of resulting graphs demonstrated confusion with the role of e and d in the plot. Specifically, it is indicated that e provides a domain tolerance and d a range tolerance.

1. Consider the "friendly" argument discussed in the lab, discuss how long the repeated selection of e-values and countering d-values must continue? What must happen in order for you to conclude that the limit as x approaches a is L? *We can get within the e of x as x approaches a by limiting values of x to achieve the desired degree of closeness to L.*

2. Define the function
 	
`> f:= proc( x )`
`x^(1/2);`
`end proc:`
	`> dePlot (f, 2, .5, 4, 1);`
	

This dePlot proves that as x approaches 4, the given e finds the distance away from 4, which is d.	

Figure 178. Group P_1's response to Lab 4, Problems 1 and 2.

Likewise, when asked, in problem 3, to demonstrate that the limit was not 1.99, i.e. $\lim_{x\to 4}\sqrt{x} \neq 1.99$, students failed to produce a relevant counterexample (see Figure 179). The graph instead suggested it *was* possible to keep the functions value within 0.5 units of 2 by keeping x within 1 unit of 4 and, thus providing support against the stated claim!

The error made in the creation of the tool related to the d-band appears in the output of problems 4 and 5 yet the pair did not observe that the intended vertical d-band is in fact misplaced. This error did not appear in Problem 3 (Figure 179) due to the particular choice of parameters in the call to `dePlot`.

3. Define the function

```
> f := proc( x )
      x^(1/2);
   end proc:
```

```
> dePlot (f, 1.99, .5, 4, 1);
```

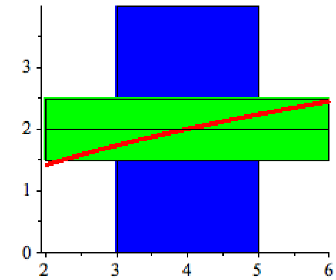


Figure 179. Group P_1's response to Lab 4, Problem 3

5. Define the function

```
> f := proc( x );
      abs(x-1)/(x-1);
   end proc:
```

```
> dePlot(f, 1, 1, .5, 1);
```

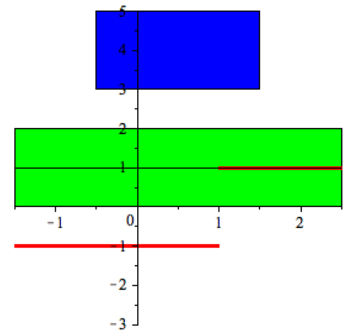


Figure 180. Group P_1's response to Lab 4, Problem 5.

In determining the greatest d-tolerance for limits for the limit statements $\lim\limits_{x \to 4} 2x - 1 = 7$ for (e=0.5), $\lim\limits_{x \to 4} \frac{x^2 - 16}{x - 4} = 8$ for (e=0.5 and e=0.05), and $\lim\limits_{x \to 0} x \sin\left(\frac{1}{x}\right)$ for (e=0.05), the pair produced one plot per problem and made no statements relating to the output. The pair's response to the limit $\lim\limits_{x \to 0} x \sin\left(\frac{1}{x}\right)$ showed they mistakenly believed the limiting value of the expression was π but gave no indication as to how they arrived at this number.

After the lab, the pair responded to post-lab questions. When presented with a `dePlot` indicating the limiting value was not 7, one member indicated this implied the

limit did not exist and the other indicated that since the true limiting value was contained within at least one of the two shaded rectangles, the limit existed. Neither proposed a limiting value.

When presented with a `dePlot` suggesting a particular limit, one of the two students was able to describe this implication but provided no further explanation, the other student thought the two plots represented independent plots of two different functions with the same limiting value.

When asked whether a `dePlot` *proved* a given limit statement, one said yes and the other no. Their answers were determined by examining the functions value at x=1 only. That is they looked at the graphs rather than the shaded rectangles provided in the plot were not considered when elaborating on their reasons for their respective statements.

Finally, when shown limit tables, as produced in Lab 3. Neither student made a connection between the columns in one-sided limit tables and the shaded rectangles in the `dePlots`. Moreover, neither student explained how the rectangles in a `dePlot` corresponded with statements in the formal definition of limit. These responses suggest little understanding of the individual domain and range processes let alone their coordination. Thus, an understanding of APOS steps 5 and 6, were not in evidence for this group.

Both students gave each other perfect peer-reviews, in spite of the fact that one of the students did a majority of the work on this last lab due to excessive absences of her partner.

Group P_2

This group correctly created the tool (see Figure 181) and demonstrated its use but did not respond to any of the questions posed in the assignment. In addressing the post-lab questions, neither students related the information in the limit tables to the `dePlot`s and provided no explanation as to how their answers were obtained. Neither student provided a response to questions relating to the rectangles in the `dePlot`s and their relation to statements in the formal definition. Thus an understanding of APOS steps 5 and 6 was not demonstrated.

```
> dePlot := proc( f, L, e, a, d)
   # declare any necessary local variables here
   #
   local p1, L1, r1, r2;

   # include graphics tools (display, and rectangle)
   #
   with( plottools ):  with( plots ):

   # plot the graph
   #
   p1 := plot( f(x), x=a-2*d..a+2*d, discont=true, thickness=2);
   L1:= line([ a-2*d, L], [ a+2*d, L], color=blue, linestyle=DASH);
   r1:= rectangle([a-2*d, L+e], [a+2*d, L-e], color=green);
   r2:= rectangle([a-d, L+4*e], [a+d, L-4*e], color=black);

    # display the graph
   display([p1,L1,r1,r2]);
   end proc:

> dePlot(f, 4, 1, 2, 1);
```

Figure 181. Group P_2's implementation of `dePlot` tool.

Group P_3 (singleton)

Like group P_2, the pair produced a correct procedure but failed to utilize it to justify any limiting behaviors requested in the lab. On the post-lab activities, the participating student indicated that if a function failed to fall within the shaded rectangles, the limit was non-existent rather than indicating the limiting value might be something different. This student was however able to correctly infer an appropriate mathematical limit statement implied by a sequence of `dePlot`s suggesting at least some understanding of the tools output. No understanding of the relation of the bounding rectangles produced by `dePlot` to statements in the formal definition was indicated. An understanding of APOS steps 5 and 6 was not in evidence.

Group P_4 (singleton)

As with the other programming groups, this group produced a correct procedure for producing the plots requested but failed to respond to any of the requested limits. On the post -lab, the participating student did not demonstrate any understanding of the plots produced by `dePlot`. As with all the prior programming groups, an understanding of APOS steps 5 and 6 was not demonstrated.

Group N_1

This group was able to properly define functions and utilize the provided `dePlot` tool. In responding to the first problem, shown in Figure 182, they indicated that the d value must change in order to ensure that the functions value remained within e of the proposed limiting value suggesting understanding of the need for coordination in the limit process. However, they did not indicate the necessity of continuing this process indefinitely so as to trap the limit.

On the second problem, the pair correctly produced plots in which they began with the suggested e and d values and systematically varied the d value so as to achieve the specified e value before reducing the e value for the next attempt thus focusing on finding appropriate domain behavior for a fixed range tolerance. This coordinated exploration is shown in Figure 183.

1. Consider the "friendly" argument discussed in the lab, discuss how long the repeated selection of e-values and countering d-values must continue? What must happen in order for you to conclude that the limit as x approaches a is L?

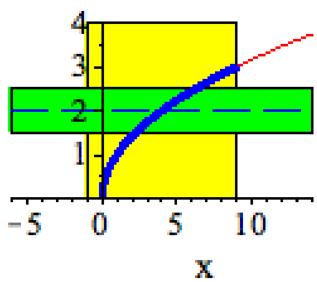
ANSWER:

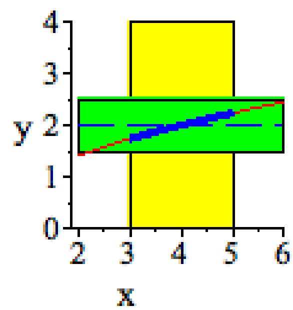
If the value of e is changed by 0.1, then d must change by .01 so that the function can exist in both the rectangles.

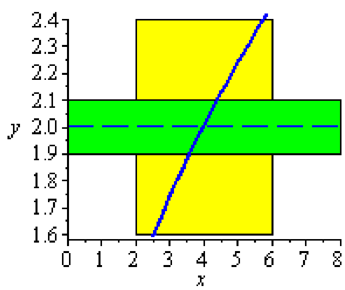
Figure 182. Group N_1's response to Lab 4, Problem 1.

<table>
<tr><td colspan="2">2. Create an argument using the <code>dePlot</code> procedure that demonstrates that
lim sqrt(x) = 2
x -> 4
Your argument must demonstrate using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

<code>> f:= x-> sqrt(x);</code></td></tr>
<tr><td>Plot #1

<code>> dePlot (f, 2, 0.5, 4, 5);</code>
</td><td>Plot #2

<code>>dePlot (f, 2, .5, 4, 1);</code>

[Desired e=0.5 tolerance achieved using d=1]</td></tr>
<tr><td>Plot #3

<code>>dePlot (f, 2, .1, 4, 2);</code>

[Next, the e=0.1 value is reduced and a suitable d value is sought.]</td><td>Plot #4

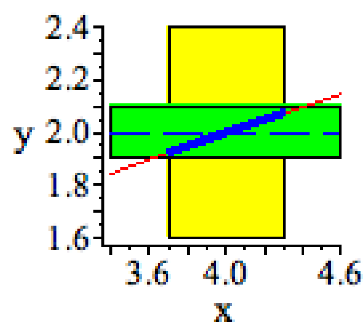
<code>> dePlot (f, 2, .1, 4, .3);</code>

[The desired e=0.1 tolerance is achieved with d=.3]</td></tr>
<tr><td colspan="2">If the value of e is changed to 0.1, then d must change by a smaller value than the original value 5, in order to for the functions to exist in both of the rectangles.</td></tr>
</table>

Figure 183. Group N_1's response to Lab 4, Problem 2

On Problem 3 and 4, the group was asked to demonstrate that a proposed limit was, in fact, not the correct limit using the tool by using the tool to produce a counterexample plot. As can be seen in the sequence shown in Figure 184, the group was able to produce a suitable counterexample and explanation. On problem three, the pair

claimed there was no limit rather than the limit was not 1.99. This distinction was however clarified in their response to Problem 4.

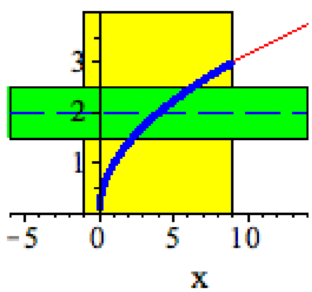
The pair appeared to understand the tools output and its interpretation, but, unlike their exploration in Problem 2 (see Figure 183), the pair tended to vary both e and d simultaneously during the exploration suggesting they might not perceive the domain and range processes as distinct but necessarily coordinated processes.

3. Using dePlot, create an argument that $\lim_{x \to 4} \sqrt{x} \neq 1.99$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

```
> f:= x-> sqrt(x);
```

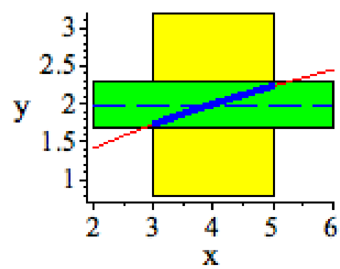
Plot #1

```
> dePlot (f, 1.99, .5, 4, 5);
```



Plot #2

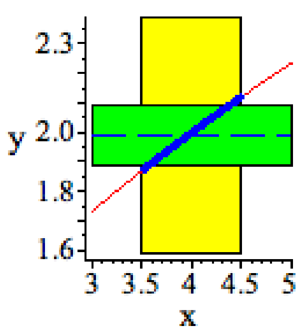
```
> dePlot (f, 1.99, .3, 4, 1);
```



[Desired e=0.3 tolerance achieved using d=1]

Plot #3

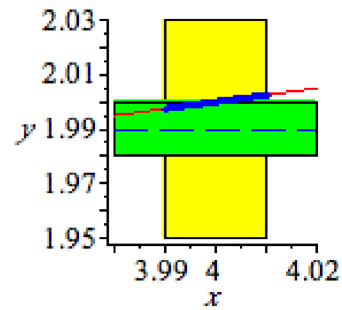
```
> dePlot (f, 1.99, .1, 4, .5);
```



[Next, the e value is reduced to 0.1 and a suitable d value is sought.]

Plot #4

```
> dePlot (f, 1.99, .01, 4, .01);
```

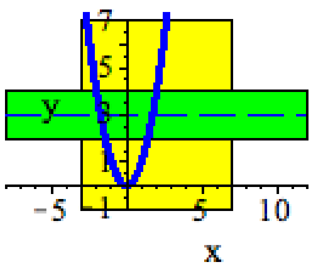
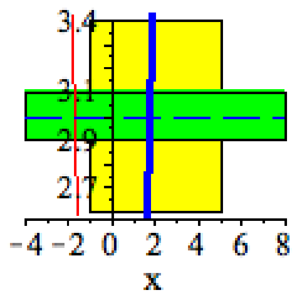


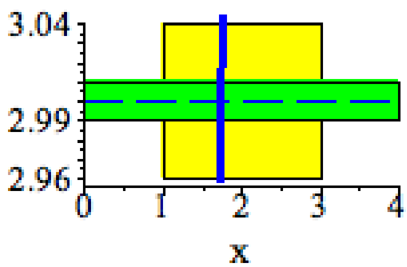
[The desired e=0.1 tolerance can not be achieved with d=.01]

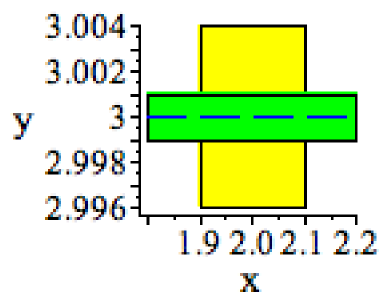
For every e value, you do not have to change the d value; therefore, the function would not exist [in the rectangles] when there is not a limit.

Figure 184. Group N_1's response to Lab 4, Problem 3

On Problem 4, Figure 185, the group mistakenly explored the limit as x approached 2 rather than 4. However, their conclusion indicates an awareness of this. As in Problem 3, the pair continued to simultaneously vary d and e to accomplish their exploration. On Problem 5, shown in Figure 186, the group correctly concluded there was no limit.

<table>
<tr><td colspan="2">4. Using dePlot, create an argument that $\lim_{x \to 4} x^2 \neq 3$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.</td></tr>
<tr><td>Plot #1
<code>> dePlot(f, 3, 1, 2, 5);</code>

</td>
<td>Plot #2
<code>>dePlot(f, 3, .1, 2, 3);</code>


[Desired e=0.5 tolerance achieved using d=1]</td></tr>
<tr><td>Plot #3
<code>>dePlot(f, 3, .01, 2, 1);</code>


[Next, the e=0.1 value is reduced and a suitable d value is sought.]</td>
<td>Plot #4
<code>> dePlot(f, 3, .001, 2, .1);</code>


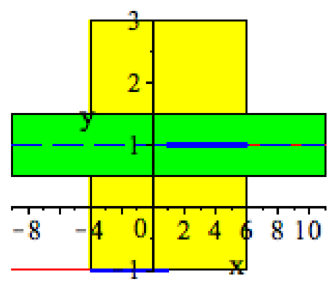
[The desired e=0.1 tolerance is achieved with d=.3]</td></tr>
<tr><td colspan="2">The limit of x^2 as x approaches 2 does not exist at 3 because the function cannot be contained within the e and d values of the graph.</td></tr>
</table>

Figure 185. Group N_1's response to Lab 4, Problem 4

5. Using dePlot, determine $\lim_{x \to 1} \frac{|x-1|}{x-1}$ or indicate that no such limit exists. In either case, your argument must show using a sequence of dePlots and corresponding written explanations that support your claim.

Plot #1

```
> dePlot(f, 1, .5, 1, 5);
```



Since the L-value is 1, the function must come closer and closer to 1 which is in the green box and the yellow box overlap. However since the e-value cannot ever be within .5 of the L-value, there is no limit at the function x=1. Therefore, the limit does not exist.

Figure 186. Group N_1's response to Lab 4, Problem 5.

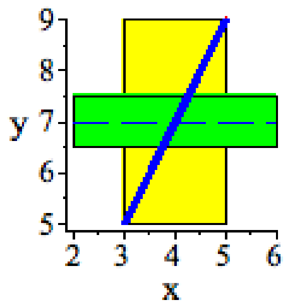
On Problem 6, the largest d-band tolerance was to be experimentally determined for a given e by utilizing the `dePlot` tool to estimate. Subsequently the estimate was verified algebraically in Problem 7. These results are shown in Figure 187. Again, rather than leaving the e value fixed at e=0.5, the pair simultaneously reduced both e and d to achieve the requested tolerance. For the e=0.05 tolerance, the pair did not show any incremental systematic exploration only a single graph with the requested tolerance achieved but rather found a the more restrictive combination e=0.01, d=0.005.

Unfortunately, although the pair did algebraically find the maximum tolerances of d for a given e tolerance, the pair did not state explicitly what these d values were nor did they make any comparison with the values they determined experimentally with the `dePlot` tool.

6. For each of the following limits, find the largest d-value that ensures that the y-values of the function are within the specified e-value of the limiting value. Find this by trial and error, producing a sequence of dePlots that lead to and support your conclusions. IN EACH PROBLEM, BEGIN WITH d=1.

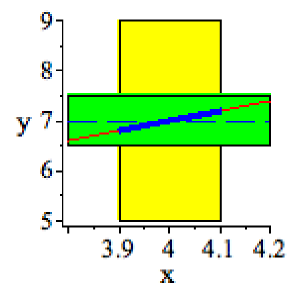
Initial Plot (6a), e=0.5

```
> dePlot( f, 7, 0.5, 4, 1);
```



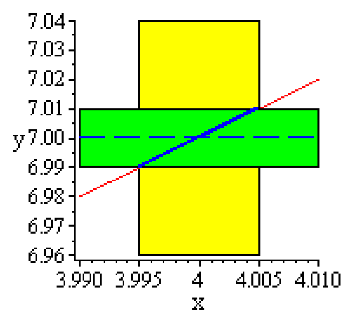
Final Plot (6a), e=0.5

```
> dePlot(f, 7, .01, 4, .005);
```



Final Plot (6b), e=0.05

```
> dePlot(f, 7, 0.01, 4, .005);
```



7. For the limit problems in question 6a and 6b, find the exact largest d-values by using Maple's solve procedure. That is, have Maple solve the equation $|f(x) - L| < e$ for x. (i.e. `solve(abs(f(x)-L)<e, x )`) and, using the range of x-values returned, determine the exact largest d-value required to keep the functions values within e of L. Show that the answers you find here are consistent with those that you found in question 6. Clearly explain how you determined the exact d-values from the solutions Maple provided.

[Algebraic Confirmation for 6a.]

```
> solve(abs(f(x)-7)<.5, x);
           RealRange(Open(3.750000000), Open(4.250000000))
```

We obtained the largest d-value by changing the range b/w which the points can fall, and seeing if the line lies in both boxes.

```
> solve(abs(f(x)-5)<.05, x );
              RealRange(Open(0.9500000), Open(1.050000))
```

Figure 187. Group N_1's response to Lab 4, Problems 6 and 7

On the post-lab, only one member submitted responses. While this student was unable to correctly interpret the `dePlot`s shown, she very clearly understands the connection between columns of limit tables and intervals in `dePlot`s as shown in Figure 188. She was one of only three students who clearly stated this connection.

The student who did not respond was the student who struggled with programming concepts. As a group, this group did demonstrate APOS Steps 5. However, individually, one member did not progress beyond understanding the coordination necessary in the domain and range processes in clearly in evidence here, APOS Step 3c. APOS Step 6 was not demonstrated as there was not a systematic coordinated variation in the selection of the e and d values. Following this lab, the student achieving only APOS Step 5 did adopt the correct formal definition of limit.

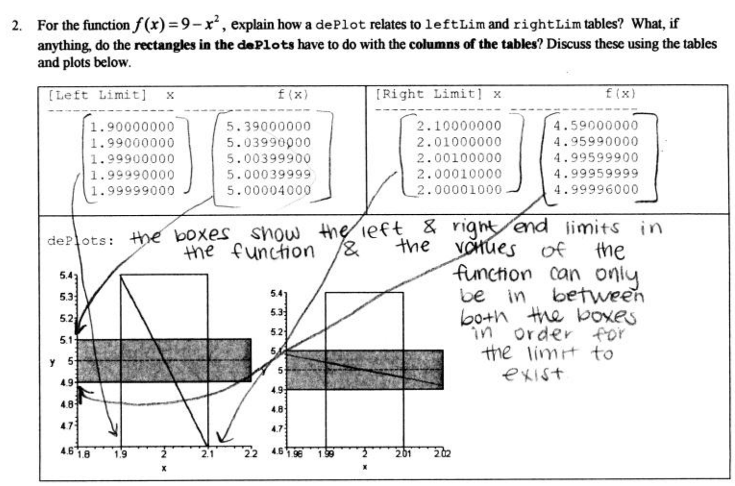


Figure 188. Group N_1's understanding of connection between columns of limit tables and rectangles in dePlots

Group N_2

This group was able to properly define functions and utilize the exploratory tool. In responding to the first problem they correctly indicated the process would continue indefinitely as shown in Figure 189.

On Problem 2, the pair correctly produced plots in which they began with the suggested e and d values and systematically varied both d and e values so as to achieve a plot in which the specified e tolerance was in fact achieved (see Figure 190). As with group N_1, the pair understood the way an exemplar plot must appear, however, the way in which the both d and e were simultaneously decreased suggests the pair does not deeply understand the individual domain and range processes. An understanding of the need for coordination is demonstrated; however there is a simultaneous lack of understanding of the independence of the two processes.

1. Consider the "friendly" argument discussed in the lab, discuss how long the repeated selection of e-values and countering d-values must continue? What must happen in order for you to conclude that the limit as x approaches a is L?

ANSWER:

Ad infinitum. Like the guy with no face in The Phantom Tollbooth, the jobs continue on forever, with no clear end in sight.

Figure 189. Group N_2's response to Lab 4, Problem 1

2. Create an argument using the dePlot procedure that demonstrates that

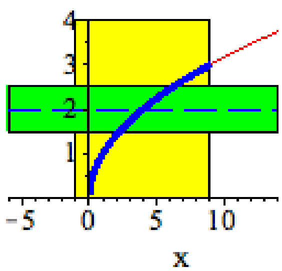
$$\lim_{x \to 4} \text{sqrt}(x) = 2$$

Your argument must demonstrate using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

```
> f:= x-> sqrt(x);
```

Plot #1

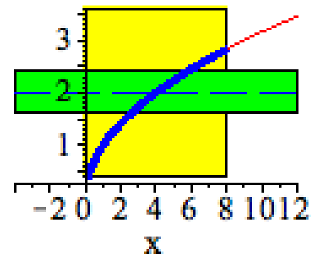
```
> dePlot (f, 2, 0.5, 4, 5);
```



This is what you said to start with.

Plot #2

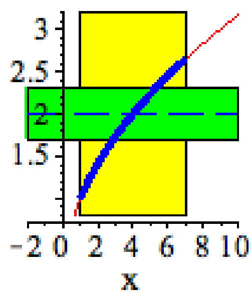
```
> dePlot( f,2,.4,4,4);
```



We have now gotten smaller by a relative unit.

Plot #3

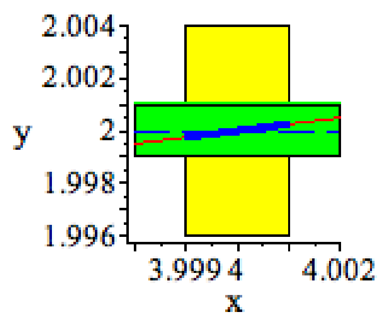
```
> dePlot( f,2,.3,4,3);
```



et cetera.

Plot #4

```
> dePlot (f, 2, .1, 4, .3);
```



et cetera, et cetera. but different.

Figure 190. Group N_2 's response to Lab 4, Problem 2

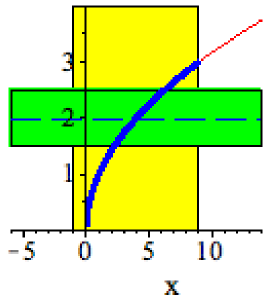
On problem three and four, see Figure 191, the group was asked to demonstrate that a proposed limit was, in fact, not the correct limit using the tool by using the tool to produce a counterexample plot. As can be seen in the sequence below, the group was able to produce suitable counterexamples but the group employed the same strategy of simultaneously decreasing d and e.

3. Using dePlot, create an argument that $\lim_{x \to 4} \sqrt{x} \neq 1.99$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

```
> f:= x-> sqrt(x);
```

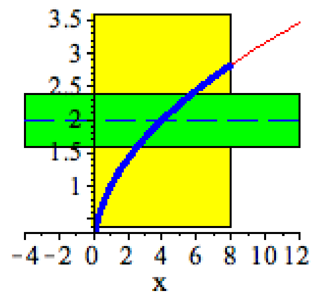
Plot #1

```
> dePlot (f, 1.99, .5, 4, 5);
```



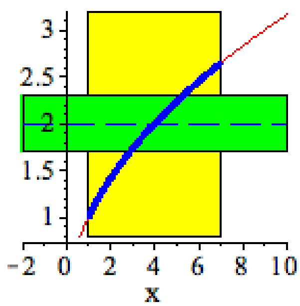
Plot #2

```
> dePlot (f, 1.99, .4, 4, 4);
```



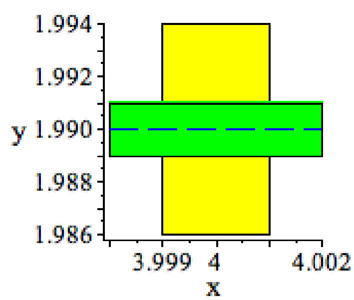
Plot #3

```
> dePlot (f, 1.99, .3, 4, 3);
```



Plot #4

```
> dePlot (f, 1.99, .001, 4,
.001);
```



See? Not even on the graph. HA.

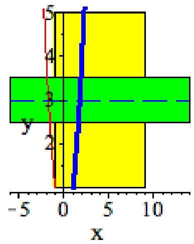
Figure 191. Group N_2's response to Lab 4, Problem 3

On Problem 4, the group was similarly successful. Their exploration is shown in Figure 192 .

4. Using dePlot, create an argument that $\lim_{x \to 4} x^2 \neq 3$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

Plot #1

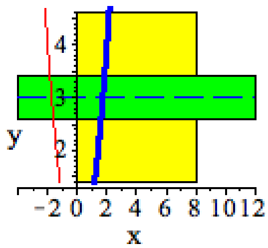
```
> dePlot(f, 3, .5, 4, 5);
```



This is what you said to start with.

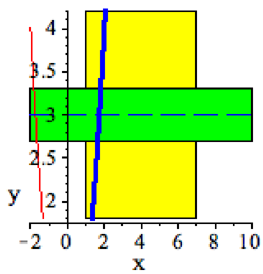
Plot #2

```
>dePlot(f, 3, .4, 2, 4);
```



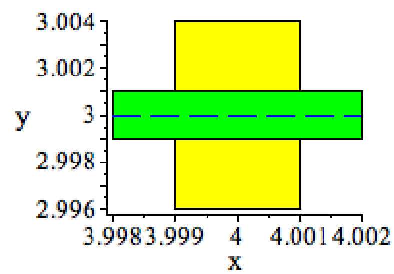
Plot #3

```
>dePlot(f, 3, .3, 4, 3);
```



Plot #4

```
> dePlot( f, 3, .001, 4, .001);
```



Once again, you can't see it, so it's not there. Good logic, huh.

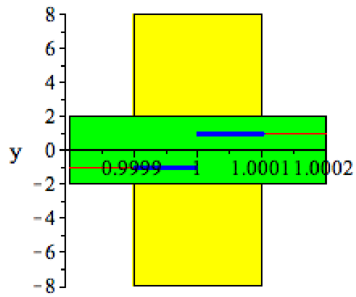
Figure 192. Group N_2's response to Lab 4, Problem 4

On Problem 5, the group produced a sequence of plots showing that in fact it was possible to achieve a tolerance of e=2 units of L=0.This is shown in Figure 193. The group did not justify, using the `dePlot`, why a limit fails to exist at x=1. Specifically, they could have indicated the right hand limit at 1 appears to be 1 and subsequently show, using the plot, that it is not possible to achieve a closeness of say e=0.5 of L=0, etc. The pair reached an accurate conclusion utilizing reasoning based upon their understanding of continuity rather than an argument based upon output from the `dePlot` tool. Additionally, the pair continued its strategy of simultaneously varying e and d.

5. Using dePlot, determine $\lim_{x\to 1} \frac{|x-1|}{x-1}$ or indicate that no such limit exists. In either case, your argument must show using a sequence of dePlots and corresponding written explanations that support your claim.

Final Plot

```
> dePlot(f,0,2,1,.0001);
```



Looks like a fractal--the closer we get, it still looks the same. This won't change any, so: Jump discontinuity at x=1. Thus, no limit as x->1.

Figure 193. Group N_2's response to Lab 4, Problem 5

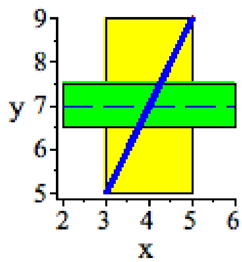
On Problem 6, the students were asked to determine the largest d-band tolerance that would ensure a given e-band tolerance and, in Problem 7, provide algebraic confirmation of these estimates using Maple. This required experimentation using the `dePlot` tool. Although the group demonstrated some confusion relating to the coordination of the domain and range processes, there responses here indicate understanding of the respective domain and range processes.

As is shown in Figure 194, the pair systematically varied the d values while maintaining the e value of 0.5 but in the end changed the e value to 0.02. They then estimated the maximum d value is for this altered e-value. Subsequently, the pair used Maple to confirm the d-range for the e-tolerance they utilized thus suggesting understanding of the requisite size of the d-range for a given e-value. A similar exploration was made for 6b, see Figure 195. No response was given for Problem 6c.

<table>
<tr><td colspan="2">6a. For each of the following limits, find the largest d-value that ensures that the y-values of the function are within the specified e-value of the limiting value. Find this by trial and error, producing a sequence of dePlots that lead to and support your conclusions. IN EACH PROBLEM, BEGIN WITH d=1.</td></tr>
<tr><td>

Initial Plot (6a), e=0.5

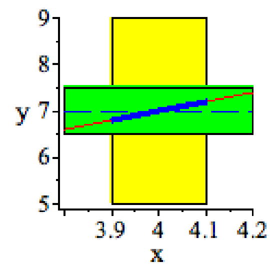
`> dePlot( f, 7, 0.5, 4, 1);`



</td><td>

Final Plot (6a), e=0.5

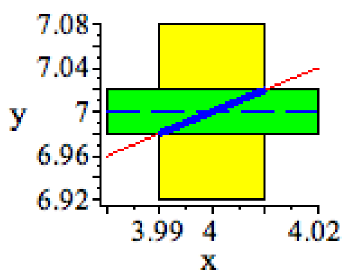
`>dePlot( f, 7, 0.5, 4, 0.1);`



</td></tr>
<tr><td>

Final Plot (6a), e=0.05

> dePlot(f, 7, 0.02, 4, 0.01);



[Note: The e value was mistakenly changed to 0.02]

</td><td>

7. Algebraic Confirmation for 6a

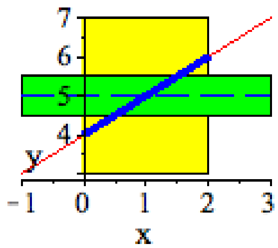
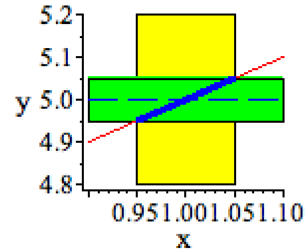
`> solve( abs( f(x)-7 ) < .02, x );`

`RealRange(Open(3.990000), Open(4.010000))`

d=.01, e=.02

</td></tr>
</table>

Figure 194. Group N_2's response to Lab 4, Problems 6a and 7.

<table>
<tr><td colspan="2">6b. For each of the following limits, find the largest d-value that ensures that the y-values of the function are within the specified e-value of the limiting value. Find this by trial and error, producing a sequence of dePlots that lead to and support your conclusions. IN EACH PROBLEM, BEGIN WITH d=1.</td></tr>
<tr><td>Initial Plot (6b)
<code>>dePlot(f, 5, .5, 1, 1);</code>

</td><td>Final Plot (6b)
<code>> dePlot(f, 3, .05, 1, .05);</code>

</td></tr>
<tr><td colspan="2">7. Algebraic Confirmation for 6b

<code>> solve(abs(f(x)-5)<.05, x);</code>
<code>RealRange(Open(0.9500000000), Open(1.050000000))</code>

d=.05, e=.05</td></tr>
</table>

Figure 195. Group N_2's response to Lab 4, Problems 6b and 7

On the post-lab activities, only one group member submitted responses. This student correctly interpreted all the `dePlot`s shown and clearly understood that no particular set of e-d tolerances would effectively prove a particular limiting value but rather that an infinite number of such plots are necessary to ultimately know the limiting value. He also elaborated that, when shown a `dePlot` demonstrating a particular limiting values was not correct, a single `dePlot` could definitively support this conclusion.

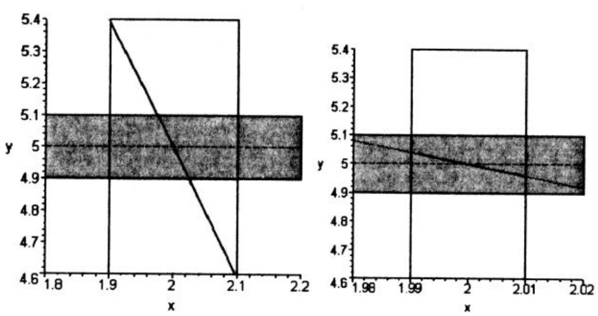
Finally, as shown in Figure 196, this student clearly understands the connection between columns of limit tables of `leftLim` and `rightLim`, and the rectangles appearing in `dePlot`s. This group has attained APOS Step 5 and demonstrated a limited partial understanding of Step 6 in their understanding based upon these results.

2. For the function $f(x) = 9 - x^2$, explain how a `dePlot` relates to `leftLim` and `rightLim` tables? What, if anything, do the **rectangles in the `dePlots`** have to do with the **columns of the tables**? Discuss these using the tables and plots below.

[Left Limit] x	f(x)
1.90000000	5.39000000
1.99000000	5.03990000
1.99900000	5.00399900
1.99990000	5.00039999
1.99999000	5.00004000

[Right Limit] x	f(x)
2.10000000	4.59000000
2.01000000	4.95990000
2.00100000	4.99599900
2.00010000	4.99959999
2.00001000	4.99996000

dePlots:



The rectangles bound the areas in the tables.

Figure 196. Group N_2's understanding of connection between columns of limit tables and rectangles in `dePlots`

Group N_3

This group was able to properly define functions and utilize the exploratory tool. As shown in their response to Problem 1, in Figure 197, they correctly indicated the process would continue indefinitely and that the limiting value might not ever be attained.

On the second problem, shown in Figure 198 , the pair correctly produced plots in which they demonstrated the requested e tolerances were attainable but did not produce a sequence of plots that reveal their manner of exploration. The group additionally considered ever decreasing values of e to ensure that they had inferred the correct limit. Notably, like the other groups, they vary e and d simultaneously so as to achieve the necessary support for their claims.

1. Consider the "friendly" argument discussed in the lab, discuss how long the repeated selection of e-values and countering d-values must continue? What must happen in order for you to conclude that the limit as x approaches a is L?

ANSWER: You must be able to make the argument last forever because with a limit you can be closer and closer without ever reaching your value.

Figure 197. Group N_3's response to Lab 4, Problem 1

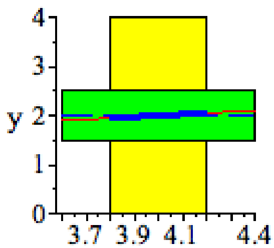
2. Create an argument using the dePlot procedure that demonstrates that $\lim_{x \to 4} \sqrt{x} = 2$

Your argument must demonstrate using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

```
> f:= x-> sqrt(x);
```

Plot #1

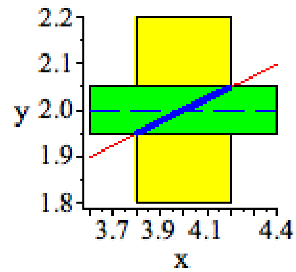
```
> dePlot (f, 2, 0.5, 4, 0.2);
```



[Shows e=0.5 is attainable.]

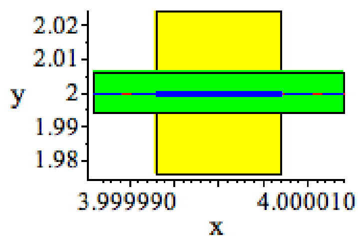
Plot #2

```
> dePlot( f,2,.0.05,4,0.2);
```



Final Plot #3

```
> dePlot( f, 2, 0.006, 4, 0.000007);
```



The limit of f is 2 proven by the dePlots above.

Figure 198. Group N_3's response to Lab 4, Problem 2

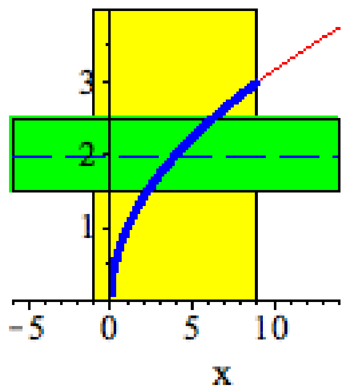
On Problem 3 and 4 (see Figure 199 and Figure 200), the group was asked to demonstrate that a proposed limit was, in fact, not the correct limit using the tool by using the tool to produce a counterexample. As can be seen in the sequence in Figure 199, the group was able to produce suitable counterexamples but the group employed the same strategy of simultaneously decreasing d and e.

3. Using dePlot, create an argument that $\lim_{x \to 4} \sqrt{x} \neq 1.99$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

```
> f:= x-> sqrt(x);
```

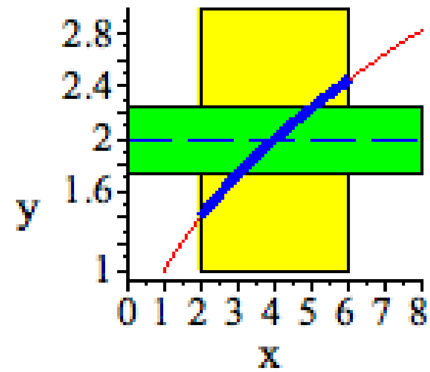
Plot #1

```
> dePlot (f, 1.99, .5, 4, 5);
```



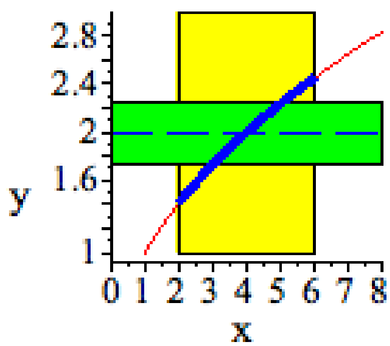
Plot #2

```
> dePlot (f, 1.99, 0.25, 4, 2);
```



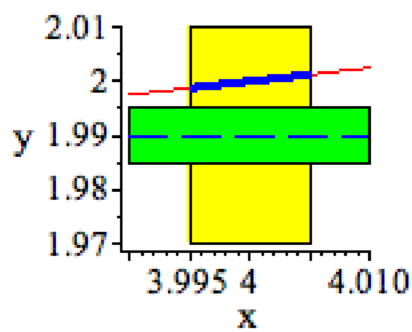
Plot #3

```
> dePlot (f, 1.99, 0.3, 4, 1);
```



Plot #4

```
> dePlot(f,1.99, 0.005, 4,
0.005);
```

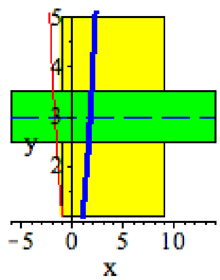
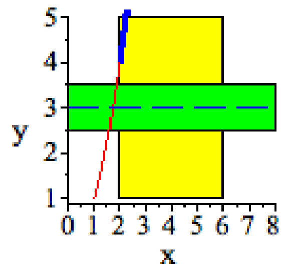
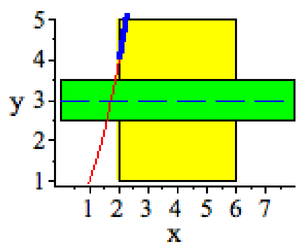
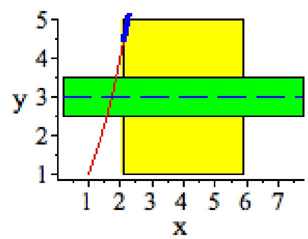


See? Not even on the graph. HA.

We got closer and closer to the point where it showed the value of the function at 4, and from the last graph you can tell that the limit of the function is not 1.99 because the line is not located within both the limits- being the green and yellow boxes.

Figure 199. Group N_3's response to Lab 4, Problem 3

It was noted that beginning with Problem 4, the pair began systematically varying parameters d and e in a way suggesting an awareness of the independence, yet coordinated-ness, of the domain and range processes. In this counterexample, the pair let e remain fixed as they varied d to achieve the requested tolerances.

<table>
<tr><td colspan="2">4. Using dePlot, create an argument that $\lim_{x \to 4} x^2 \neq 3$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.</td></tr>
<tr><td>Plot #1
<code>> dePlot(f, 3, .5, 4, 5);</code>

</td><td>Plot #2
<code>>dePlot(f, 3, 0.5, 4, 2);</code>

</td></tr>
<tr><td>Plot #3
<code>>dePlot(f, 3, 0.5, 4, 1.99);</code>

</td><td>Plot #4
<code>> dePlot(f, 3, 0.5, 4, 1.89);</code>


This shows that the limit as x goes to 4 is not 3 because the line does not pass through both the boxes at the same interval.</td></tr>
</table>

Figure 200. Group N_3's response to Lab 4, Problem 4

On Problem 5, see Figure 201, the group produced a sequence of plots showing very clearly there was no limit at x=1. The reasoning they used was their understanding of continuity and, most significantly, the output of the `dePlot` tool. Their comments clearly indicate not only why L=1 could not be limit but that no limit at x=1 can exist.

As shown in Figure 202, Figure 203, and Figure 204, problem six, the pair was asked to determine the largest d-band tolerances that would ensure a given e-band tolerance. Subsequently, in Problem 7, they were to use Maple to confirm these tolerances algebraically. This required experimentation using the `dePlot` tool. On problem 6a, the group found a tolerance for e=0.05 but did not do so for e=0.5.

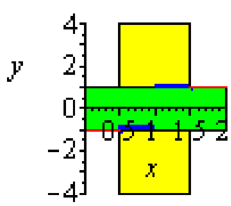
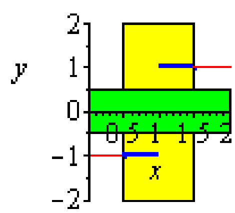
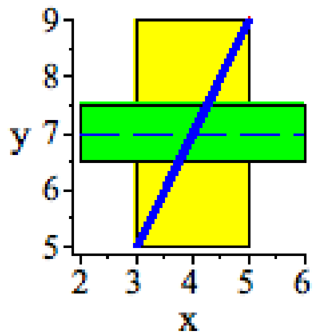
<table>
<tr><td colspan="2">5. Using dePlot, determine $\lim_{x\to 1}\frac{|x-1|}{x-1}$ or indicate that no such limit exists. In either case, your argument must show using a sequence of dePlots and corresponding written explanations that support your claim.</td></tr>
<tr><td>Initial Plot, e=1, d=0.5
<code>> dePlot(h,0,1,1,.5);</code>

</td><td>Final Plot, e=0.5, d=0.5
<code>> dePlot(h,0,.5,1,.5);</code>

</td></tr>
<tr><td colspan="2">The limit does not exist because at the point x=1 the function is not continuous. No matter how narrow we make the yellow [vertical] rectangle, the graph is outside the green [horizontal] rectangle.</td></tr>
</table>

Figure 201. Group N_3's response to Lab 4, Problem 5

6a. For each of the following limits, find the largest d-value that ensures that the y-values of the function are within the specified e-value of the limiting value. Find this by trial and error, producing a sequence of dePlots that lead to and support your conclusions.

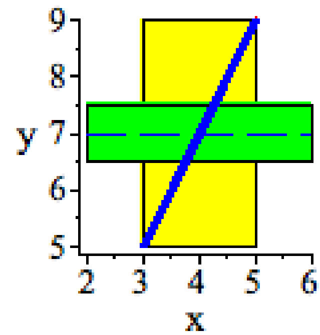
Initial Plot (6a), e=0.5

```
> dePlot( f, 7, 0.5, 4, 1);
```



Final Plot (6a), e=0.5

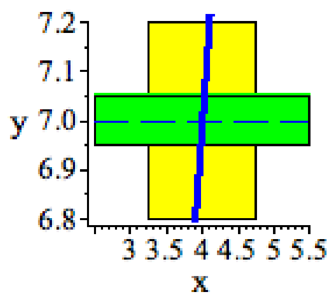
```
>dePlot( f, 7, 0.5, 4, 1);
```



[pair did not vary d in any way]

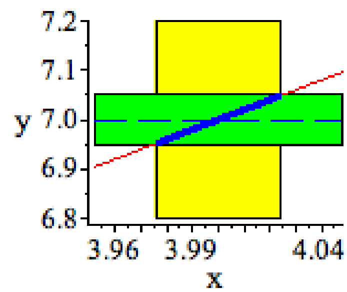
Initial Plot (6a), e=0.05

```
> dePlot( f, 7, 0.05, 4, 0.75);
```



Final Plot (6a), e=0.05

```
> dePlot( f, 7, 0.05, 4, 0.024);
```



We concluded from the information from our guess and check method above that the largest d value possible is .024 or somewhere very close to this.

7. Algebraic Confirmation of 6a

```
> solve( abs( f(x)-7 ) < 0.05, x );

                RealRange(Open(3.750000000), Open(4.0250000000))
```

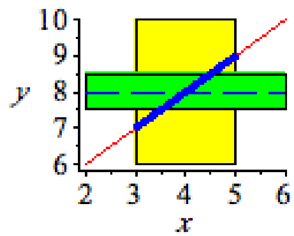
This shows the limit is 7 as x approaches 4 because the value 4 for x is included in the range as the functions output value gets closer to 7.

Figure 202. Group N_3's response to Lab 4, Problem 6a and 7

6b. For each of the following limits, find the largest d-value that ensures that the y-values of the function are within the specified e-value of the limiting value. Find this by trial and error, producing a sequence of dePlots that lead to and support your conclusions.

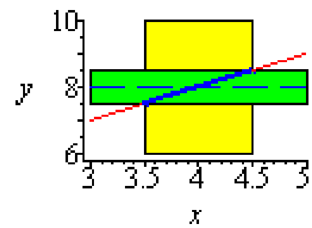
Initial Plot (6b), e=0.5

```
>dePlot( f, 8, 0.5, 4, 1);
```



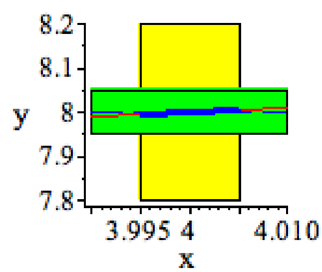
Final Plot (6b), e=0.5

```
> dePlot( f, 8, 0.5, 4, 0.5);
```



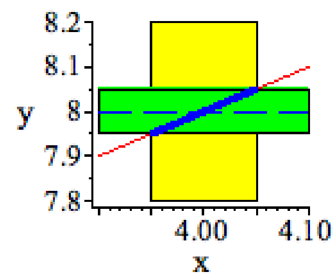
Initial Plot (6b), e=0.05

```
>dePlot( f, 8, 0.05, 4, 0.005);
```



Final Plot (6b), e=0.05

```
> dePlot( f, 3, .05, 1, .05);
```



We concluded from the information from our guess and check method above that the largest d value possible is .05 or somewhere very close to this.

7. Algebraic Confirmation for question 6b

```
> solve(abs(n(x)-8)<0.5);
RealRange(Open(4.), Open(4.500000000)), RealRange(Open(3.500000000),
Open(4.))
```

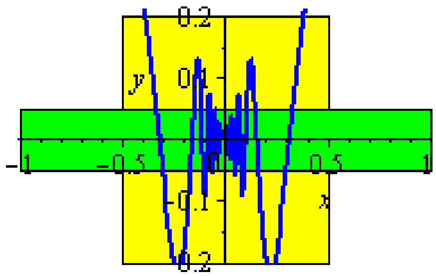
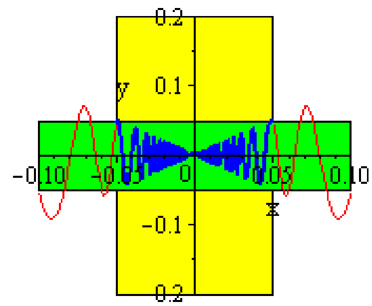
d=.5, e=.5

```
> solve(abs(n(x)-8)<0.05);
RealRange(Open(4.), Open(4.050000000)), RealRange(Open(3.950000000),
Open(4.))
```

d=.05, e=.05

This shows as the function value goes to 8, the input for that value is getting closer to this interval. The value 4 is both in this interval and equidistant from the endpoints of this interval, showing that the limit of the value 4 is 8.

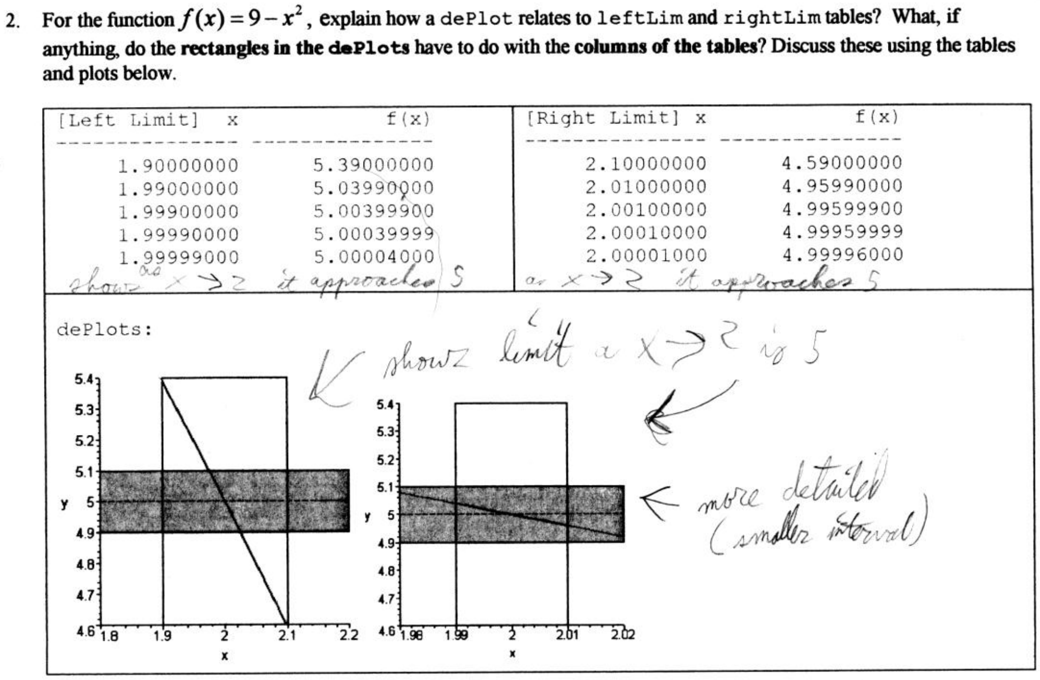
Figure 203. Group N_3's response to Lab 4, Problem 6b and 7

<table>
<tr><td colspan="2">6c. For each of the following limits, find the largest d-value that ensures that the y-values of the function are within the specified e-value of the limiting value. Find this by trial and error, producing a sequence of dePlots that lead to and support your conclusions.</td></tr>
<tr><td>Initial Plot (6c), e=0.05
<code>> dePlot(m,0,.05,0,.5);</code>

</td><td>Final Plot (6c), e=0.05
<code>> dePlot(m,0,.05,0,.05);</code>

</td></tr>
<tr><td colspan="2">We concluded from the information from our guess and check method above that the largest d value possible is .05 or somewhere very close to this.</td></tr>
<tr><td colspan="2">7. Algebraic Confirmation for question 6c

[No algebraic justification was provided!]</td></tr>
</table>

Figure 204. Group N_3's response to Lab 4, Problem 6c and 7

On the post-lab questions, both students understood that a single `dePlot` could provide a counterexample to a proposed limiting value. However, one group member, understood the process would need to continue indefinitely in order to prove a proposed limiting value. This student very clearly understood the connection between the columns of limit tables and the rectangles in `dePlot`s as shown in Figure 205.



2. For the function $f(x) = 9 - x^2$, explain how a `dePlot` relates to `leftLim` and `rightLim` tables? What, if anything, do the **rectangles in the `dePlots`** have to do with the **columns of the tables**? Discuss these using the tables and plots below.

[Left Limit] x	f(x)	[Right Limit] x	f(x)
1.90000000	5.39000000	2.10000000	4.59000000
1.99000000	5.03990000	2.01000000	4.95990000
1.99900000	5.00399900	2.00100000	4.99599900
1.99990000	5.00039999	2.00010000	4.99959999
1.99999000	5.00004000	2.00001000	4.99996000

shows as $x \to 2$ it approaches 5

as $x \to 2$ it approaches 5

dePlots:

Figure 205. Group N_3's members understanding of the connection between the columns in limit tables and the rectangles in `dePlots`

The other member, the student having extensive prior programming experience, understands the necessity for simultaneously bounding the limit point within two bounding rectangles but did not see the necessity for an infinite process.

This group very clearly understands the coordination between the domain and range processes as evidenced by the systematic methods of parameter variation discovered and utilized during this lab exercise. As a group has attained APOS level 5. Of note is that one member attained APOS level 6.

Group N_4

This group was able to properly define functions and utilize the exploratory tool. In responding to the first problem they correctly indicated the process would continue indefinitely, see Figure 206.

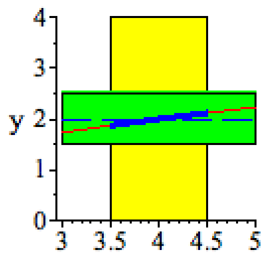
On the second problem, the pair correctly produced plots in which they demonstrated the requested e tolerance was attainable. The group varied e and d simultaneously as most other groups (see Figure 207). Of note, is the comment that these graphs suggested but do not prove the correct limit has been found.

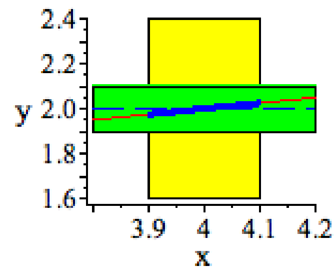
1. Consider the "friendly" argument discussed in the lab, discuss how long the repeated selection of e-values and countering d-values must continue? What must happen in order for you to conclude that the limit as x approaches a is L? *ANSWER: The line must be in the overlap area for the given x and y range. You can always and forever continue making different intervals for the graph to plot in. You can be mildly satisfied when the line is in both domains and never more, never more, out of the Domain.*

Figure 206. Group N_4's response to Lab 4, Problem 1

<table>
<tr><td colspan="2">2. Create an argument using the dePlot procedure that demonstrates that $\lim_{x\to 4}\sqrt{x}=2$
Your argument must demonstrate using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.

<code>> f:= x-> sqrt(x);</code></td></tr>
<tr><td>Plot #1

<code>> dePlot (f, 2, 0.5, 4, 0.5);</code>

[Shows e=0.5 is attainable.]</td><td>Plot #2

<code>> dePlot(f,2,.0.0001,4,0.0001);</code>
</td></tr>
<tr><td>Final Plot #3

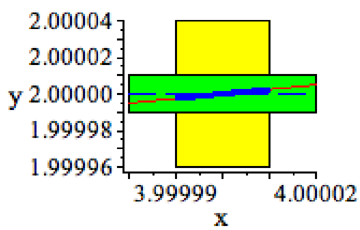
<code>> dePlot(f,2,0.00001,4,0.00001);</code>
</td><td></td></tr>
<tr><td colspan="2">After analyzing the plots above, we have decided that the limit as x approaches 4 is 2, even though you can not completely tell from these graphs.</td></tr>
</table>

Figure 207. Group N_4's response to Lab 4, Problem 2

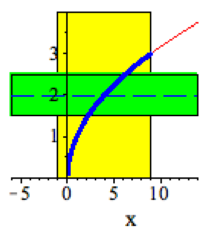
On Problems 3 and 4, the group was asked to demonstrate that a proposed limit was, in fact, not the correct limit using the tool to produce counterexample plots (see Figure 208 and Figure 209). As can be seen in the sequence of plots, the group was able to produce suitable counterexamples and in doing so employed a systematic strategy of varying d for a fixed e tolerance suggesting deeper understanding of the coordinated domain and range processes. Interestingly, the group did not feel that this constituted proof. In the prior problem, the pair correctly indicated that the `dePlot` procedure could

not be utilized to absolutely prove that a given limit was correct. In this problem, the group fails to recognize that the tool can be utilized to absolutely disprove a given limiting value. Suggesting the pair has a misunderstanding of the nature of mathematical proof. On Problem 4, the group mistakenly explored the limit as x approached 2 rather than 4. However, their conclusion indicates an awareness of this.

3. Using dePlot, create an argument that $\lim_{x\to 4}\sqrt{x} \neq 1.99$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.
> f:= x-> sqrt(x);

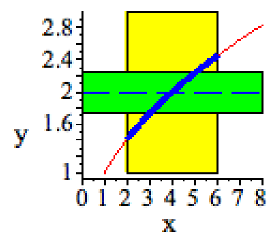
Plot #1

```
> dePlot (f, 1.99, .5, 4, 5);
```



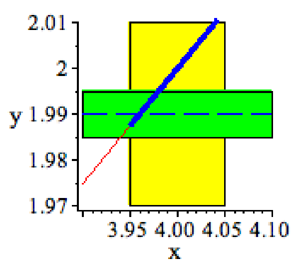
Plot #2

```
> dePlot (f,1.99,0.005,4,1);
```



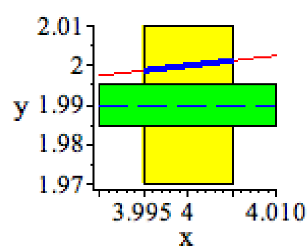
Plot #3

```
> dePlot (f, 1.99, 0.005, 4,
0.05);
```



Plot #4

```
> dePlot (f,1.99,0.005,4,0.005);
```



The graph of the dePlots show that when a small Domain is used to contain the Limit, one can tell that the plot of 4 for x does not yield a 1.99 for y. Using the dePlot tool it is impossible to prove the statement one way or the other.

Figure 208. Group N_4's response to Lab 4, Problem 3

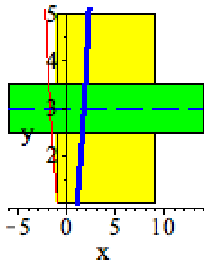
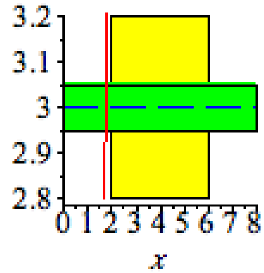
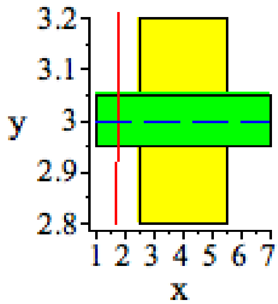
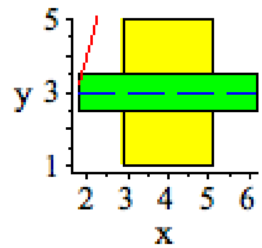
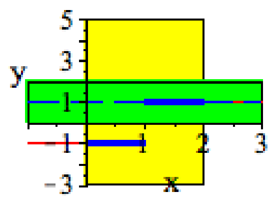
4. Using dePlot, create an argument that $\lim_{x \to 4} x^2 \neq 3$. Your argument must show using a sequence of at least 4 dePlots and written explanations that support your claims. Begin with e=0.5 and d=5.	
Plot #1 `> dePlot(f, 3, 0.5, 4, 5);` 	Plot #2 `>dePlot(f, 3, 0.05, 4, 2);` 
Plot #3 `>dePlot(f, 3, 0.05, 4, 1.5);` 	Plot #4 `> dePlot( f, 3, 0.05, 4, 1.1);` 
As you can see from the dePlots above, the limit of does not equal 3 because the graph does not intersect the 3 in the minimal domain we imposed.	

Figure 209. Group N_4's response to Lab 4, Problem 4

On Problem 5, see Figure 210 , the group produced a sequence of plots showing that in fact it was possible to achieve a tolerance of e=1 units of L=1. This group made the key observation that 1 could not be the limit as one approached x=1, they indicated the right hand limit at 1 appears to be 1 and additionally showed, using the plot, that it is not possible to achieve a closeness of e=0.001 of L=1, etc. In addition to the use of the `dePlot` tool, that pair reasoned using their understanding of continuity to further support their claim.

<table>
<tr><td colspan="2">5. Using dePlot, determine $\lim_{x\to 1}\frac{|x-1|}{x-1}$ or indicate that no such limit exists. In either case, your argument must show using a sequence of dePlots and corresponding written explanations that support your claim.</td></tr>
<tr><td>Initial Plot

<code>> dePlot(f, 1, 1, 1, 1);</code>

If Maple is used, (1,1) is not true in the dePlot because 1 cannot be the limit with a discontinuous jump.</td>
<td>Final Plot

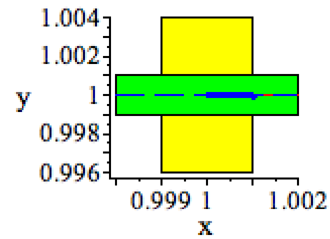
<code>> dePlot(f, 1, 0.001, 1, 0.001);</code>

The dePlot shows that the positive value yields a (1,1) intersection, but the negative 1 value does not intersect, so in order to work there would have to be a x>1 boundary, but still the function is not continuous and contains a jump, therefore there is not a limit to the function.</td></tr>
</table>

Figure 210. Group N_4's response to Lab 4, Problem 5

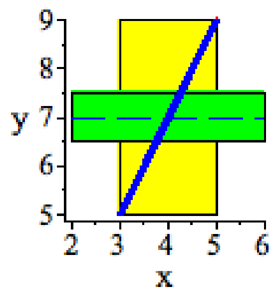
On Problem 6, see Figure 211 and Figure 212, the pair was asked to determine the largest d-band tolerance that would ensure a given e-band tolerance. Subsequently, in Problem 7, they were unable to have Maple confirm this estimate algebraically in 6a because they thought the limiting value was 6 according to their input. Additionally, in part 6b, the pair does not parenthesize the function properly so that Maple does not return a suitable interval. Interestingly, the pair does not seem to be conflicted between the results they find and those that Maple produced.

The pair utilized a coordinated domain and range process to explore the limit and succeeded in finding a d tolerance that achieves the desired e tolerance, but was unable to determine the largest such value suggesting the pair did not understand how to determine this characteristic utilizing the `dePlot` tool. The pair did not find a tolerance for e=0.5 but did so for e=0.05. Notably, the group did systematically vary the d value for a fixed e value.

6. For each of the following limits, find the largest d-value that ensures that the y-values of the function are within the specified e-value of the limiting value. Find this by trial and error, producing a sequence of dePlots that lead to and support your conclusions.

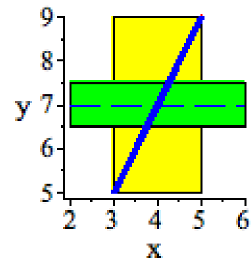
Initial Plot (6a), e=0.5

```
> dePlot( f, 7, 0.5, 4, 1);
```



Final Plot (6a), e=0.5

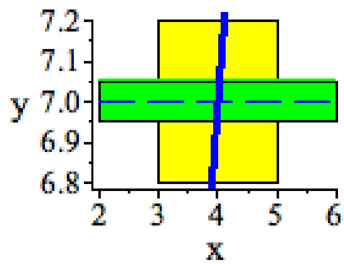
```
>dePlot( f, 7, 0.5, 4, 1);
```




```
[pair did not vary d]
```

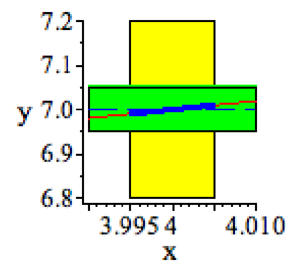
Initial Plot (6a), e=0.05

```
> dePlot( f, 7, 0.05, 4, 1);
```



Final Plot (6a), e=0.05

```
> dePlot( f, 7, 0.05, 4, 0.005);
```



We will conclude that due to the above dePlot, that (4,7) exists as a Limit in the function, all points land within the domain.

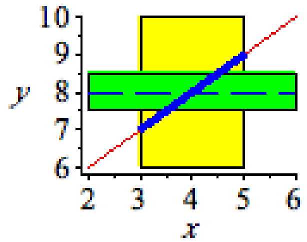
Algebraic Confirmation (problem 7)

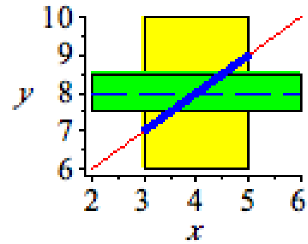
```
> solve( abs( 2*x-5) < 2 );
                RealRange(Open(3/2), Open(7/2))
```

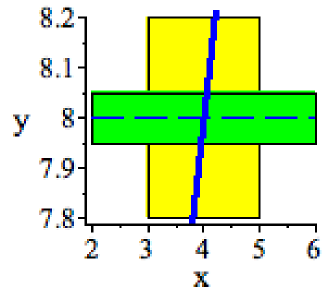
The boundary on the x line shows that the limit exists at 2.5 or a and the d range is 1, it's the largest d range available to close in the function in the l range of 5 and e range of 2.

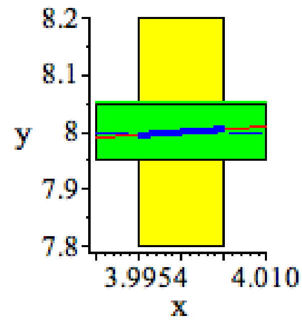
Figure 211. Group N_4's response to Lab 4, Problem 6a, 7

<table>
<tr>
<td>Initial Plot (6a), e=0.5

<code>>dePlot(f, 8, 0.5, 4, 1);</code>
</td>
<td>Final Plot (6ab), e=0.5

<code>> dePlot(f, 8, 0.5, 4, 1);</code>

[Pair did not vary d for e=0.5]</td>
</tr>
<tr>
<td>Initial Plot (6b), e=0.05

<code>>dePlot(f, 8, 0.05, 4, 1);</code>
</td>
<td>Final Plot (6b), e=0.05

<code>> dePlot(f, 8, .05, 1, .005);</code>

We will conclude that (4,8) is the limit point in this function due to the function remaining in the entire domain created in the dePlot.</td>
</tr>
<tr>
<td colspan="2">Algebraic Confirmation (problem 7)

<code>> solve(abs(x^2 - 16 / x - 4) < 0.5, x);</code>
<code>RealRange(Open(2.978547575), Open(3.106518887))</code>

The boundary on the x line shows that the limit exists at an of estimated 3.1, l of 2.97. Largest boundaries are shown.</td>
</tr>
</table>

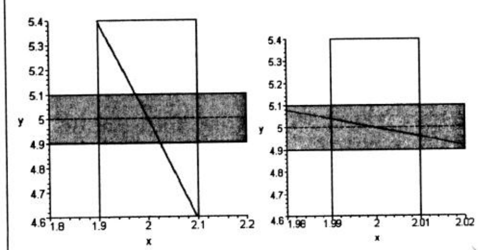
Figure 212. Group N_4's response to Lab 4, Problem 6b, 7

On the post-lab questions, both students understood that a single `dePlot` could provide a counterexample to a proposed limiting value. As with group N_3, one understood the process would needs to continue indefinitely in order to prove a proposed limiting value. This student further understood the connection between the columns of limit tables and the rectangles in `dePlots` as shown in the upper response of Figure 213.

2. For the function $f(x) = 9 - x^2$, explain how a `dePlot` relates to `leftLim` and `rightLim` tables? What, if anything, do the **rectangles in the `dePlots`** have to do with the **columns of the tables**? Discuss these using the tables and plots below.

[Left Limit] x	f(x)	[Right Limit] x	f(x)
1.90000000	5.39000000	2.10000000	4.59000000
1.99000000	5.03990000	2.01000000	4.95990000
1.99900000	5.00399900	2.00100000	4.99599900
1.99990000	5.00039999	2.00010000	4.99959999
1.99999000	5.00004000	2.00001000	4.99996000

dePlots:

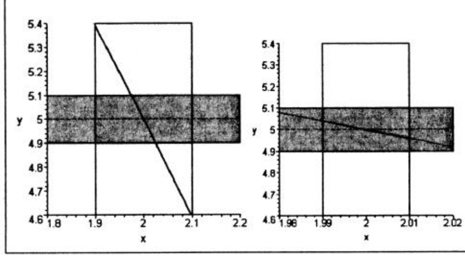


The dePlot graphs are visual reperesentations of the left and right limit tables.
The horizontal retectangles corespand to the x column and the verticle rectangle to the f(x) column.

2. For the function $f(x) = 9 - x^2$, explain how a `dePlot` relates to `leftLim` and `rightLim` tables? What, if anything, do the **rectangles in the `dePlots`** have to do with the **columns of the tables**? Discuss these using the tables and plots below.

[Left Limit] x	f(x)	[Right Limit] x	f(x)
1.90000000	5.39000000	2.10000000	4.59000000
1.99000000	5.03990000	2.01000000	4.95990000
1.99900000	5.00399900	2.00100000	4.99599900
1.99990000	5.00039999	2.00010000	4.99959999
1.99999000	5.00004000	2.00001000	4.99996000

dePlots:



The x inputs vary b/w 1.9 and 2.1 so they create an x range of [1.9, 2.1].
The y inputs vary b/w 5.1 and 4.9 so they create an y range of [4.9, 5.1].

Figure 213. Group N_4's understanding of connection between columns of limit tables and rectangles in `dePlots`

Finally, both students understand the coordination necessary between the domain and range processes, APOS level 3c. Additionally, one of the pair understands the necessity of an infinite sequence of `dePlot`s in order to determine a limit. Based upon their coordinated exploration of using the tool and their postlab responses, one student progressed to APOS Step 5, the other to APOS Step 6.

Lab 4 Summary

None of the programming groups were able to successfully utilize the tool. As with prior lab exercises, this class continues to perceive the development of the tools as the task. Of the programming groups, the most successful group, P_2, failed to make relevant use of the tools in this final lab.

In stark contrast, all the non-programming groups were able to make significant progress toward the final APOS goal of understanding the formal definition of limit. Although as groups, none of the non-programming groups of class N progressed beyond APOS Step 5, each group possessed at least one student who did progress to APOS Step 6. As will be seen, these are the individuals who ultimately selected the correct formal definition. These attainment are summarized in Figure 214.

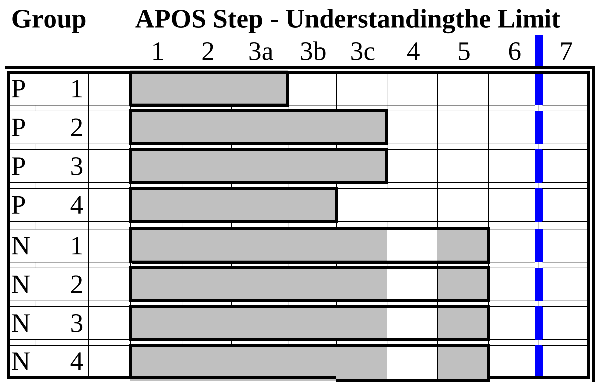


Figure 214. Final APOS Level of Attainment. The shaded region represents actual attainment and the vertical line indicates the intended level of attainment.

Final Limit Conceptions

Following this final lab, students were asked on the final exam to select the correct formal definition of limit. None of the programming groups selected the proper conception as shown in Figure 215 whereas all the non-programming groups had at least one member selecting the correct formal conception.

Group		Limit Conception
P_1	H	1
	H	2
P_2	H	4
	L	4
P_3	H	4
P_4	H	2
N_1	L	3
	L	X
N_2	H	1
	L	3
N_3	H	6
	L	3
N_4	L	6
	L	3

Conceptions of Limit (Williams, 1991)

1) (*Dynamic-theoretical*) A limit describes how a function moves as x moves towards a certain point.
2) (*Acting as a boundary*) A limit is a number or point past which a function cannot go.
3) (*Formal*) A limit is a number that the y-values of a function can be made arbitrarily close to by restricting x-values.
4) (*Unreachable*) A limit is a number or point the function gets close to but never reaches.
5) (*Acting as an approximation*) A limit is an approximation that can be made as accurate as you wish.
6) (*Dynamic-practical*) A limit is determined by plugging in numbers closer and closer to a given number until the limit is reached.

H = Higher-performing, L = Lower-performing, X = no response

Figure 215. Final Conceptions of Limit

CHAPTER FIVE

CONCLUSIONS

The purpose of this study was to investigate, characterize, and contrast the development of student understanding of the limit and derivative concepts as cultivated using pre-developed tools versus using student-developed (constructionist) programming-based tools.

Following a typical introductory instructional sequence introducing the limit concept, students were surveyed to determine their initial conceptions of limit. Subsequently, the development of understanding was chronicled and analyzed using laboratory assignments refining their initial conceptions toward the formal definition of limit. This exploration was directed toward the essential research questions,

(1) How are students' conceptual understandings different or similar having utilized programming based activities as contrasted with non-programming based activities in the CAS environment and

(2) How do these differences and similarities arise?

This chapter summarizes and contrasts results presented in Chapter 4 by comparing the varying limit conceptions and levels of attainment within the APOS framework of classes P and N, and by considering likely themes influencing these similarities and differences. Finally, implementation challenges encountered during the study are explored and suggestions for future research suggested.

Conceptions of the Limit Concept

The notion of limit is particularly difficult for students due to its dual nature of process and object. Students are generally comfortable with the idea of a mathematical object such as a number or an algebraic expression. Similarly, they are familiar with a concept of a process producing an object as in the steps in solving an equation or of performing long division. What is particularly difficult about the limit process is that it is a process that does not terminate with the production of an object, rather it is a process that is entraps the resulting object, the limit. That students struggled with these ideas is evidenced by the continuous variation in their conception of limit over the duration of the study shown in Figure 216. The shaded items indicate selection of the correct formal definition of limit at the conclusion of the study.

Group		**Initial**	**Post lab 3**	**Final**
P_1	H	1	1	1
	H	1	2	2
P_2	H	1	4	4
	L	3	4	4
P_3	H	2	4	4
P_4	H	2	3	2
N_1	L	4	1	3
	L	1	1	X
N_2	H	1	1	1
	L	2	1	3
N_3	H	1	1	6
	L	1	1	3
N_4	L	2	6	6
	L	1	1	3

Conceptions of Limit (Williams, 1991)

1) (*Dynamic-theoretical*) A limit describes how a function moves as x moves towards a certain point.
2) (*Acting as a boundary*) A limit is a number or point past which a function cannot go.
3) (*Formal*) A limit is a number that the y-values of a function can be made arbitrarily close to by restricting x-values.
4) (*Unreachable*) A limit is a number or point the function gets close to but never reaches.
5) (*Acting as an approximation*) A limit is an approximation that can be made as accurate as you wish.
6) (*Dynamic-practical*) A limit is determined by plugging in numbers closer and closer to a given number until the limit is reached.

H = Higher-performing, L = Lower-performing,
X = no response

Figure 216. Changes in Limit Conception for all Groups

The Programming Groups Final Conception of Limit

The programming groups, P_i, concluded the study with the general consensus that a limit is "(4) *a number or point the function gets close to but never reaches*." and as "(2) *a number or point past which a function cannot go.*" To these students, the limit is seen as an *unreachable boundary*.

The Non-programming Groups Final Conception of Limit

In contrast, the non-programming groups, N_i, had a much more dynamic perspective. To them the limit was perceived to be "(3) *a number that the y-values of a function can be made arbitrarily close to by restricting x-values*" and as being "(6) *determined by plugging in numbers closer and closer to a given number until the limit is reached*." These groups see the limit as an *attainable and computable* (via entrapment) *value*.

Variations in Limit Concept

The variations in conceptual understanding were observed over the course of four laboratory activities. Prior to the first lab activity, students were surveyed to determine their initial conception(s) of limit. The two most common notions expressed by both groups P and N were, first, that a limit was static entity, a "(2) *value serving as a boundary past which the functions value may not go*" and, second, that a limit described "(1) *the way a functions moves as a particular x-value is approached*." Thus, the limit was seen as a static entity assumedly produced by respective strategies introduced in the initial instructional sequence. Not surprisingly, these conceptions naturally derive from

this traditional instruction sequence utilizing analysis of graphs and the algebraic techniques of substitution, factoring, and the conjugate method.

Lab 1 specifically addressed Steps 1 and 2 of the APOS decomposition. All groups attained the intended action level of understanding of function evaluation as well as an ability to perform a sequence of coordinated evaluations. Additionally, the lab provided participants the opportunity to make inferences regarding limiting behavior utilizing sequential arguments by fostering an interest in systematic domain processes. Unfortunately, five of the eight groups, $P_2, P_4, N_1, N_3,$ and N_4 , failed to make use of readily available data to construct sequential arguments for limiting behavior. Instead, these groups favored graphical and algebraic arguments based upon understandings developed during the preliminary instructional sequence. The remaining two groups P_1 and N_2 did offer valid sequential arguments. Of the two classes, P and N, the non-programming groups seemed less willing to offer such sequential support; however, at this point in the study, there was not much differentiation among the groups in instructional terms.

As the study progressed and students began to utilize (develop and utilize, respectively) specifically designed computational tools and the pair-programming model of interaction, these conceptions began to change. In Lab 2, attention was focused specifically upon developing understanding of APOS steps 1, 2, 3a, and 3b for understanding the limit. Steps 3a and 3b were of particular interest as they required application of sequential argumentation.

Recall, students were reluctant to make sequential arguments in the first lab, tending to rely upon graphical and algebraic methods. To address this, a collection of

mysterious functions were presented to the participants for analysis. The mysterious nature of these functions was due to deliberate representational restrictions placed upon them. Specifically, students could only evaluate the prescribed functions and did not have access to either graphical or algebraic representations. The purpose of these restrictions was to prohibit the groups' ability to use graphical and algebraic arguments to justify limiting behavior and favor the use of sequential arguments.

All the programming groups of class P were able to successfully create the `simpleLimitTable` tool and demonstrate its use. However, none of the programming groups successfully explored the specified functions using the tool suggesting they may have perceived the construction of the tool as the task rather than the exploration of the given functions. Groups in class N, on the other hand, all attained the desired level of understanding within the APOS decomposition and actually utilized the tool to justify the behavior of the mystery functions.

It is with this lab the two classes, P and N, begin to diverge as evidenced by their differing levels of attainment within the APOS framework and their differing conceptual perspectives. At the conclusion of Lab 2, every non-programming group had offered at least one sequential argument supporting a limiting trend. Only one of the programming groups P_1 had offered such an argument. This appears to be an important step in the developmental path to the formal definition. Sequential understanding appears to necessarily precede the adoption of the correct formal definition.

In Lab 3, the focus was on the coordination of the domain and range processes, APOS Step 3c of the APOS decomposition, via the construction and use of two tools, `leftLim` and `rightLim`. Groups were given the opportunity to create a suitable

domain process, APOS Step 3a, and an understanding of the resulting range process, APOS Step 3b, through the design, construction, and use of functions having prescribed limiting behavior. Groups were required to support these constructed behaviors using the `leftLim` and `rightLim` tools.

Programming groups, $P_1, P_3,$ and P_4, failed to constructively utilize the `leftLim` and `rightLim` procedures although all groups were successful with their implementation. These three groups all mistakenly defined an alternative function that squared its' input and attempted to analyze it rather than the intended mystery functions. Apparently, this was due to confusion with prelab activities. Prior to the lab, the implementation of the squaring function was used as an example of procedural function definition with the programming groups. Subsequently, these groups failed to understand they were to analyze the intended mystery functions- the ones I could not entice them to explore in Lab 2. Group P_2 was the only programming group to effectively utilize the tools by applying them to the intended functions. Group P_3 appeared to be able to interpret the output from the tools but either did not apply it to the correct function or was unable to implement the requested behaviors due to programming difficulties. Beyond confusion relating to which functions were to be analyzed, even when participants were able to utilize the tool, the P groups rarely made correct inferences using the resulting output.

It appeared the P groups were cognitively overloaded by the combination of having to implement and utilize the two tools. This confusion obscured the intended goals of the lab leading participants to see the creation of the tool as the end rather than the analysis of the given mystery functions. Groups P_2 and P_3 were the only programming

groups able to correctly interpret output from the tools and correspondingly were the only programming groups to achieve an understanding of APOS Step 3c, coordination between the domain and range processes.

The non-programming groups of class N were far more successful in their analysis and application of the `leftLim` and `rightLim` tools. These groups had a clearer idea of what was being requested in the lab as well as how this information was to be discerned. Groups N_2 and N_3 were both highly successful on this lab being able to analyze, create, and justify the behavior of the provided functions using the tools. Group N_4 was similarly successful with the analysis of the mystery functions but experienced some difficulty in the construction of functions possessing requested behaviors. Group N_1 did not understand how to use the tool and, as a result, could not produce relevant data from which to draw conclusions. Of particular interest was the observation that both groups N_2 and N_3 utilized the tools to help synthesize functions b,c,d, and e; they utilized the tool to create and design functions possessing requested behavior rather than simply as a tool of analysis. These were the only two pairs in either class N or P to utilize the tool in this manner.

At the conclusion of Lab 3, the consensus among students in the programming groups, P_i, was that (4) *a limit is a number or a point the function gets close to but never reaches*. In the non-programming groups, N_i, there was general agreement that (1) *a limit described how a function moves as x move toward a particular value*. Of note is the non-programming group's abandonment of the notion of a limit as a boundary conception in favor of a more dynamic perception. As previously observed, *every* student who eventually adopted the correct formal definition adopted this dynamic-theoretical

perspective, choice (1), along their developmental path to the formal definition.

In Lab 4, none of the programming groups in class P were able to successfully utilize the tool; these groups continued to perceive the development of the tools as the task. Of the programming groups, the most successful group, P_2, failed to make relevant use of the tools in this final lab. In contrast, all the non-programming groups were able to make significant progress toward the final APOS goal of understanding the formal definition of limit. Although as groups, none of the non-programming groups of class N progressed beyond APOS Step 5, each group possessed at least one student who did progress to APOS Step 6. As will be seen, these are the individuals who ultimately selected the correct formal definition.

At the close of the study, the overall consensus among all groups was that the limit was (3) *a number that the y-value of a function can be made arbitrarily close to by restricting the x-values*- the correct formal conception, or that the limit was (4) *a number past which a function gets close but never reaches*. Of particular interest is that these two conceptions were functions of the groups!

Students in the non-programming groups, N_i, tended to adopt the correct formal definition and students in the programming groups, P_i, indicated the limit was an unreachable boundary, two very different conceptions of limit.

Interestingly, there were two programming groups that contained members who, at some point, selected the correct formal definition and later rejected it. A member of group P_2 initially selected the correct definition but later decided to adopt the limit is a number that a function get close too but never reaches. Group P_4 selected the correct

formal definition following Lab 3 but ultimately rejected in favor of choice 2. It is not clear why this occurred.

Levels of APOS Attainment

In terms of the APOS framework for understanding limits, the groups of classes N and P had varying degrees of success in reaching the conceptual objectives put forth as shown in Figure 217. It should be noted that no attempt to specifically address APOS Step 4 was made in the study as I was unable to devise an activity of appropriate difficulty.

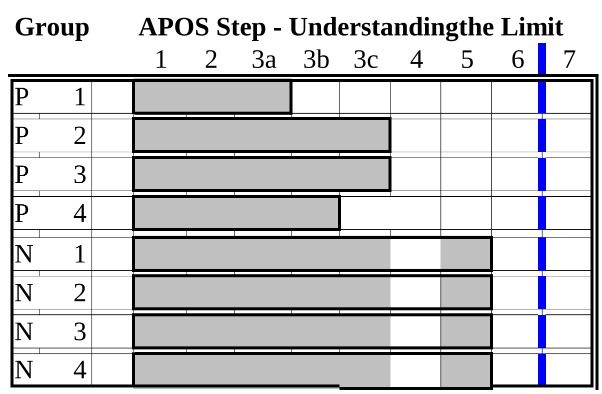


Figure 217. Final APOS Level of Attainment. The shaded region represents actual attainment and the vertical line indicates the intended level of attainment.

Ultimately, all non-programming groups contained a member who achieved Steps 5 and 6 in the APOS framework for limits. Groups P_2 and P_3 progressed through Step 3c in the framework and the remaining groups, P_1 and P_4, progressed at best through APOS Step 3b.

Understanding Step 3c appears to be the most important component in the APOS decomposition. The groups that ultimately selected the correct formal definition, N_1, N_2, N_3 and N_4, all progressed to this step at the very least. Students who did not adopt the correct formal definition seem to have fallen into a trap in which they perceive the limit as an unattainable boundary. It appears that once a student adopts this perspective, it is very difficult to change it. Students who selected perspective (2) and/or (4) never seem to change it *unless they develop the requisite sequential perspective*.

The non-programming groups ultimately had the greatest success with all four of the four groups, $N_1, N_2, N_3,$ and N_4, each having a member adopting the correct formal definition of limit although no individual or group was ultimately able to express, in writing, the formal definition of limit. Group N_1 experienced difficulty using the tool in Lab 3. As a result, as a group, their APOS progression stopped at Step 3b. Interestingly, in the fourth lab, one member of this group was able to effectively utilize the tool and, as a result, came to understand and adopt the correct formal definition ultimately accomplishing APOS Step 6.

Group P_1 progressed least of any of the programming groups despite having two high performing females as members. The two members worked well together, however, one member was frequently absent creating additional burden on the remaining student as well as creating discontinuities.

Sources of Differences in Conceptual Understanding

Beyond the themes specifically outlined by the APOS decomposition for understanding limit, four themes appeared to correlate with the observed trends in the development of conceptual understanding in this study: (a) an *instructional focus* on skills rather than concepts (at least in terms of understanding limits), (b) the *instructional sequence* utilized to introduce the limit concept, (c) the extent of each group's *adoption and utilization of the computational* tools and (d) the *ways in which conceptual conflict was mediated within each group*.

The Role of the Instructional Focus

When students are first introduced to algebraic methods for finding limits, such as the factoring and subsequence cancellation of factors in rational expressions, this tends to reinforce the limit as an object produced by a finite sequence of steps. As with many modern calculus texts, arguments involving subtle sequential (process-based) reasoning are rarely addressed or glossed over in the course text. For example, the classic and highly informative example is the computation of the limit, $\lim_{x \to 0}\left(\frac{\sin(x)}{x}\right)$. This particular example is relegated to an appendix in the course text and is generally seen as too abstract for inclusion in a beginning calculus course.

When the concept of limit is graphically formulated, it again appears to be a finite sequence of steps, i.e. look where the two sides of the functions graph meet (or fail to meet) at a given point. Students are not conscious of any underlying infinite processes as such processes are inconveniently masked and hidden from introspection by the human visual processing system.

This algebraic and graphical skills based instructional approach teaches a certain kind of mathematical knowledge that comprises skills solving isolated problems quickly, and that implicitly devalues the importance of *procedural* understanding or, put another way, of developing an appreciation of underlying mathematical concept of limit. This means that students do not appreciate the need for consistency and rigor, so do not notice conflicts, and therefore cannot learn from them. This constraining natural of this instructional approach was noted by Henri Lebesque with respect to training students for mathematical competitions.

> Unfortunately competitive examinations often encourage deception. The teachers must train their students to answer little fragmentary questions quite well, and they give them model answers that are often veritable masterpieces and that leave no room for criticism. To achieve this, the teachers isolate each question from the whole of mathematics and create for this question alone a perfect language without bothering about its relationships to other questions. Mathematics is no longer a monument but a heap. - Henri Lebesgue (Lebesque, 1966)

As previously described, students first introduced to limits algebraically and graphically develop an object conception of limit that appears *static*. Recall that initially the two most common concepts of limit were one, "A limit describes how a function moves as x moves towards a *certain point*," and two, "A limit is a number or *point past which a function cannot go.*" The limit being the point, a *static* entity, past which the function "may not move."

While this initial limit conception provides a useful intuitive notion, this study suggests it may not be a good cognitive root leading to the development of a formal

understanding of the limit. This strategy appears to lead to cognitive conflict requiring major cognitive reconstruction to understanding limits in dynamic set-theoretic terms in place of earlier static algorithmic notions. The reconstruction into coordinated sequential terms appears difficult for students.

However, while sequential (process based) understanding is crucial, there are challenges associated with fostering this understanding. Recall that two of the programming group rejected the correct formal definition after accepting it. It appears these choices may be related to the fact that in constructing the limit table procedures, students in the programming groups were more aware of the domain process as a finite process. To model the actual *infinite* domain process, *finite* looping constructs were utilized. Since any practical looping structure will necessarily be finite, students almost certainly saw this process as incapable of ever producing a single limiting value. This tended to reinforce limit as an unreachable boundary. In contrast, students in the non-programming groups could see the limiting behavior produced by the tool but they were not consciously aware of the finite nature of the process producing the tables making their focus the limit as object rather than limit as process. The limit process was likely seen as one in which they were in control by virtue of being able to personally vary the parameters to the tools as they wished whereas programming pairs creating looping constructs perceived the finite loops as being in control. It appears there is interplay between the limit as process and limit as object concepts that must be carefully addressed. Specifically, there are extents to which each model, process versus object, should be exposed to students as their limit concept develops.

The Role of the Instructional Sequence

In addition to this instructional focus, the sequence in which certain concepts are adopted appears to be important in developing the correct formal definition of limit. The most successful groups demonstrated a willingness and ability to offer sequential justification for limiting behavior. Groups $N_2, N_3,$ and N_4 continued to utilize such arguments once they were adopted whereas groups that adopted such a perspective and later abandoned it were ultimately unsuccessful. Group N_1 also offered sequential arguments in lab one and two but when they experienced an inability to properly utilize the tool in lab three, they understandably stopped providing sequential (let alone any) justification for a functions behavior. In particular, groups $P_1, P_3,$ and P_4 began utilizing such arguments in the second lab but stopped utilizing this mode of analysis in subsequent labs- ultimately they abandoned the correct formal definition.

It appears that to be successful, students must abandon notions two and four and adopt option one at some point along their developmental path. As demonstrated by most of the programming groups, once an individual made the decision to choose understanding four, "(4) the *limit is a number a function gets close to but never reaches*," students in both groups almost never abandoned this perspective. Suggesting that either it is very difficult to alter this perspective or, at the very least, that the activities utilized here are ineffective at promoting this transition. Likewise, every student who eventually adopted the correct formal definition, in groups $N_1, N_2, N_3,$ and N_4, adopted the dynamic-theoretical perspective option one.

Thus in addition to the instructional focus, the instructional *sequence* used by students to learn concepts appears significant. If students are introduced to limiting

behavior using *sequential explorations* prior to being exposed to the more traditional (and procedural) methods of evaluating limits, they may develop the necessary sequential understanding sooner and not fall into the seeming trap of seeing the limit as unreachable.

Finding an appropriate instructional sequence that minimizes the amount of cognitive reconstruction is of great importance. While creating cognitive conflict is a powerful instructional tool, too much conflict cripples and defeats the student, as in the programming groups, and too little does not promote conceptual understanding. Computational tools, such as those used by the non-programming group, appear to be a highly effective and enjoyable tool for promoting such understanding when provided to students.

The Role of Computational Tool Adoption

One key observation that pervades both groups is that students were often reluctant to utilize a new tool to justify their answers. Despite repeated attempts to prompt them to make use of the tools, students tend not to.

On Lab 2, all the programming groups except P_4 failed to state any conclusions after having correctly developed the `simpleLimitTable` tool. Even the non-programming groups demonstrated some hesitancy to use the tool. Group N_1 could not determine how the tool operated and required direct instruction as to how the tool functioned. Group N_2 made use of the tool but continued to perform individual evaluations in addition to using the tool- as if they did feel the output of the tool was reliable.

Success in adopting the key sequential understanding appears coupled with the adoption and use of the computation tools of this study. Students in the programming

group had difficulty adopting the tool due to an inability to use the tool. Students in the non-programming groups had a much easier time adopting the tool and those groups most successful made ample use of the tool.

On Lab 3, groups P_1, P_4 and N_1 were unable to draw accurate conclusions using output from the tool and group P_4 was unable make inferences about the mystery functions due to an inability to properly use the tool. The most successful groups were those who utilized to tool routinely in the construction of the requested functions and subsequent justifications of their limiting behavior, groups N_2, N_3, N_4 and P_2.

Of particular interest is the observation that students in groups N_2 and N_3 developed much deeper instrumental relationships with the computational tools as evidence by the frequency of application of the tools in Lab 3. Students in other groups, at best, would appropriately utilize the tool to *justify* limiting behavior, whereas, beginning with Lab 3, these pairs were observed using the tools to help *synthesize* desired limiting behavior. These students experimented with the tool using it to compute and investigate while other pairs only used the tools to support conjectures first worked out on paper. This ability to utilize tools for not only for support but also for exploration and synthesis seems highly relevant to the ultimate level of conceptual understanding attained.

The Role of Pair-Programming

In this study, the pair-programming model of interaction was used to foster an environment in which cognitive conflicts could be mediated with a partner. This mode of interaction appears to have played a significant role in the success of the non-programming groups.

In both groups, the pair-programming collaborative model was a definite positive in terms of increasing student engagement and enjoyment. It was very well received with students as reflected in their peer reviews as well as in casual impromptu comments to the instructor; Students looked forward to lab sessions.

Beyond engagement and enjoyment, students in both groups elicited more confidence in their work and understanding. These effects appeared to result from the exchange of talent and resources that occurs as a result of cooperation - and also the emotional support provided by collaboration. Members demonstrated a sense of belonging to a group rather than as individuals working alone.

Dysfunctional behavior was minimized because their actions (or inactions) were visible to a peer. Students were less likely to make poor choices when their peer was watching as they feared being perceived negatively. Whereas, when I intermittently observerved the group's interactions, such behaviors could easily be hidden until I left.

One interesting observation regarding this model of interaction is discordant with the traditional view of pair-programming, however. Pair-programming assigns distinguished roles to group members where the *driver* is responsible for implementation details involving the actual construction and use of the tool and the *navigator* being responsible for dealing with broader higher-level strategic (cognitive) issues. There is an explicit differentiation in levels of abstraction between partners. In this study, however, it was observed that the most effective pairs tended not to think on different levels in spite of their differing roles.

Two of the non-programming groups, N_2 and N_3, tended not to think on different levels of abstractions while interacting whereas in most other groups the peers operated

on different abstraction levels as outlined in the pair programming framework. While pair programming tended to help the programming students maintain focus on the programming task, in terms of mathematical understanding, they were not focused on the same level! This suggests that having the right tools provided to students can focus (or distract) their attention to the mathematical issues at hand rather than the programming issues.

Another significant observation was that as individual labs progressed, peer rotation toward the end of the labs decreased the effectiveness of some groups. Group N_2, had the tendency to change roles more frequently than as prescribed by the pair-programming paradigm. It appears that it may be advantageous to have these well-defined and distinct roles in force initially to help push students towards a more uniform level of common understanding but as the groups near completion, i.e. as the students reach the same cognitive level of understanding, permitting them to keep their respective roles appears to improve efficiency and, more importantly, continuity for the students..

Implementation Challenges

This study's design posed several challenges to me and the participants. First, the initial goal of studying both the limit and the derivative proved to be too ambitious. Once a week labs did not create sufficient continuity and demanded students reorient themselves to the problems after an extended absence. It became necessary to schedule labs over two or more days at a time so as to provide necessary continuity. This additional time constraints made exploring the derivative concept impossible within the semester time frame.

From my perspective, the experimental design, which appropriately sought to

provide anonymity to participants, made data collection difficult. Not knowing the participants made collecting concise log entries focusing on consenting groups difficult. In the end, much of the collected data was unusable as it pertained to students not consenting to participating in the study. If I knew which groups were participating, data collection could have focused directly on those groups resulting in deeper and more directly relevant observations. Additionally, having eight groups from which to simultaneously collect observations, answer questions, and address technical issues was challenging balancing act for me during a class period.

From the participants perspective, when a student's partner was absent an additional burden was placed on the remaining peer. The student had to work independently, without the benefit of peer interaction and discussion. Then upon their peer's return, they were burdened with trying to orient the returning peer. These problems adversely affected group P_1 during the last two labs, Lab 3 and Lab 4. In spite of being composed of two higher performing students, the groups produced few results on the final lab.

Finally, the peer review system did not function well. Most of the time peers simply gave each other perfect peer reviews based upon personal factors rather than their performance on the lab activities. Also, since there was minimal impact upon their course grade, students often did not perceive the labs as required classroom activities and as such did not undertake them as aggressively as desired. Peer review in the workplace, where there are professional consequences, can stress personal responsibility in a way that was not possible here.

Future Research Possibilities

As with any large study, more questions arise than are answered. The study suggests that computational tools such as theseb can be an effective way to promote understanding of the limit process as every non-programming group contained a member adopting the correct formal definition. Among, the most important issues seem to be allowing time for tools to be adopted and understood. When a tool is not clearly understood, exploration becomes at best, more challenging, and at worst, impossible as student attention is redirected to the tool rather than the desired object of study (e.g. All programming groups, and group N_1's difficulty with Lab 2).

The programming groups' attention was focused too extensively on the task of developing the tool that there was not time to learn to effectively use the tool. Specifically, the programming task seems to have overshadowed conceptual understanding; The chain of conceptual construction was made excessively long requiring the understanding of too many sub-concepts or programming concepts which hindered the conceptualization of the limit concept as conceptual difficulties (often related to programming rather than the limit concept) mounted along the way preventing the ascent to the peak of understanding. Non-programming groups had more success with the tools as they met the limit concept by making focused use of the tools.

Interestingly, the two most successful non-programming groups, N_2 and N_3, contained members who had at least some prior programming experience. Undoubtedly, this prior understanding of functions and parameters to functions only added to their comfort level in using the provided tools making it more likely that the tools would be used productively. Perhaps the programming strategy utilized in this study would be as

effective or more effective if programming skills were developed prior to the development and use of these tools for mathematical discovery. If students began the study with knowledge of programming, would they have the same level of success (or greater) as the non-programming groups in this study?

The original study intended to explore students understanding of the derivative as well as their understanding of the limit. As the study developed, it became clear that the original agenda was too aggressive- more time was needed to construct and utilize the computational tools. Future research could address students' conception of the derivative in light of the findings of this study.

Finally, with respect to the pair-programming model of interaction, the most successful groups tended interact in a manner contradicting a key assumption of the pair-programming paradigm. These members consistently blurred the distinction between driver and navigator by simultaneously sharing these roles during laboratory activities. This lead to the observation that, in doing this, group members were functioning on identical cognitive levels leading to higher levels of conceptual attainment for the pair. This suggests more care might be taken in creating pairs so as to best elicit this behavior. Future reseach could address how suitable pairings are best obtained.

Conclusions

As computational tools become ever more sophisticated and available, underlying mathematical concepts often become more and more obscure and hidden. Developing conceptual understanding of mathematics using technological tools will become more challenging. Helping students understanding how to build, revise and evaluate mathematical models will continue to be a primary focus in the mathematics classroom.

While traditional computational tools may hide underlying details, when properly designed, they may instead highlight key conceptual understandings. The challenge is knowing “what to hide” and “what not to hide.”

Students in the programming group had a formidable task of learning basic programming constructs coupled with subsequent utilization of the tools they developed. This proved to be too ambitious as most students devoted the majority of their time learning the programming and creating the tool. This left little time for them to learn to use the tool in any productive fashion. In most cases the students in the programming group were able to complete the programming tasks but this monopolized their time; these students viewed the programming as the task rather than as the tool. The question is whether, the mere development of the tools in any way added or detracted from their understanding of the limit concept. In this study, the answer is that they clearly detracted from their understanding.

Students in the non-programming group did a much better job utilizing the exploratory tool although they too had initial difficulty using the tool as an exploratory tool often opting to utilize more familiar algebraic tools to determine limits.

This study provides a step toward understanding how computation tools can designed and utilized to promote mathematical understanding. The interplay between these two groups speaks to the issue of what to hide and what not to hide. The programming groups were exposed to too many concepts not closely aligned with the underlying mathematical concept of limit. An approach where carefully designed tools are provided to students that specifically highlight the underlying mathematical concept being addresses can be quite effective.

References

Asiala, M., Brown, A., DeVries, D., Dubinsky, E., Mathews, D., & Thomas, K. (1996). A framework for research and curriculum development in undergraduate mathematics education. *Research in Collegiate Mathematics Education, 2,* 1-32.

Asiala, M., Cottrill, J., Dubinsky, E., & Schwingendorf, K. (1997). The development of students' graphical understanding of the derivative. *The Journal of Mathematical Behavior, 16*(4), 399-431.

Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (2 ed.): Addison-Wesley Professional.

Berry, J. S., & Nyman, M. A. (2003). Promoting students' graphical understanding of the calculus. *Journal of Mathematical Behavior*, *22*, 481-497.

Blume, G. W., & Schoen, H. L. (1988). Mathematical problem-solving performance of eight-grade programmers and nonprogrammers. *Journal for Research in Mathematics Education*, *19*(2), 142-156.

Boyatzis, R. E. (1998). *Transforming Qualitative Information*. Thousand Oaks: Sage Publications.

Clements, D. H. (1999). The future of educational computing research: the case of computer programming. *Information Technology in Childhood Education, 1999,* 147-179.

Clements, D. H., & Battista, M. T. (1989). Learning Geometric concepts in a Logo environment. *Journal for Research in Mathematics Education, 20*(5), 450-467.

Clements, D. H., & Battista, M. T. (1990). The effects of Logo on Children's conceptualizations of angle and polygons. *Journal for Research in Mathematics Education, 21*(5), 356-371.

Clements, D. H., & Meredith, J. S. (1993). Research on Logo: effects and efficacy. *Journal of Computing in Childhood Education, 4*(3-4), 263-290.

Clements, D. H., & Sarama, J. (1995). Design of a logo environment for elementary geometry. *Journal of Mathematical Behavior, 14*(4), 381-398.

Clements, D. H., & Sarama, J. (1997). Research on Logo: A decade of Progress. *Computers in the Schools, 14*(1-2), 9-46.

Cooley, L. A. (1995). *Evaluating the effects on conceptual understanding and achievement of enhancing an introductory calculus course with a computer algebra system* Unpublished Doctoral Dissertation, New York University, New York.

Cornu, B. (1991). Limits. In D. Tall (Ed.), *Advanced Mathematical Thinking* (pp. 133-166). Dordrecht: Kluwer.

Cotrill, J., Dubinsky, E., Nichols, D., Schwingendorf, K., Thomas, K., & Vidakovic, D. (1996). Understanding the Limit Concept: Beginning with a Coordinated Process Scheme. *Journal of Mathematical Behavior, 15*, 167-192.

Creswell, J. W. (1998). *Qualitative Inquiry and Research Design*. Thousand Oaks, CA.: Sage Publishing.

Creswell, J. W. (2003). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (2 ed.). Thousand Oaks, CA: Sage Publications.

Crowe, D., & Zand, H. (2000). Computers and Undergraduate Mathematics I: Setting the Scene. *Computers & Education, 35*, 95-121.

Crowe, D., & Zand, H. (2001). Computers and Undergraduate Mathematics 2: On the Desktop. *Computers & Education, 37*, 317-344.

Cunningham, R. F. (1991). *The effects on achievement of using computer software to reduce hand-generated symbolic manipulation in freshman calculus.* Unpublished Ed.D. Dissertation, Temple University.

Davis, R. B., & Maher, C. A. (1997). *How Students Think: the role of representations.* Mahwah, NJ.: Erlbaum.

Demana, F., & Waits, B. K. (2000). Calculators in Mathematics Teaching and Learning - Past, Present, and Future. *NCTM Yearbook 2000*.

DiSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy*. Cambridge: The MIT Press.

DiSessa, A., Hoyles, C., Noss, R., & Edwards, L. D. (1995). Computers and Exploratory Learning: Setting the Scene. In A. DiSessa, C. Hoyles, R. Noss & L. D. Edwards (Eds.), *Computers and Exploratory Learning* (pp. 1-12). New York: Springer.

DiSessa, A. A., Abelson, H., & Ploger, D. (1991). An Overview of Boxer. *Journal of Mathematical Behavior*, *10*(1), 3-15.

Dubinsky, E. (2000). Meaning and Formalism in Mathematics. *International Journal of Computers for Mathematical Learning, 5*(3), 211-240.

Eisenberg, M. (1995). Programmable applications: interpreter meets interface. *ACM SIGCHI Bulletin*, *27*(2), 68-93.

Erlandson, D. A., Harris, E. L., Skipper, B. L., & Allen, S. D. (1993). *Doing Naturalistic Inquiry: A Guide to Methods*: Sage Publications.

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1970). Programming-Languages as a Conceptual Framework for Teaching Mathematics. *Interfact, 4*(2), 13-17.

Fey, J. T., Cuoco, A., Kieran, C., McMullin, L., & Zbiek, R. M. (Eds.). (2003). *Computer Algebra Systems in secondary school mathematics education*. Reston, VA: National Council of Techers of Mathematics.

Flores, A. (1985). *Effect of Computer Programming on the Learning of Calculus Concepts.* Unpublished Doctoral Dissertation, The Ohio State University.

Graham, A. (2003). Creating calculator programs. *Micro Math, 19*(3), 4-43.

Gray, E., & Tall, D. (1994). Duality, Ambiguity, and Flexibility: A "Proceptual" View of Simple Arithmetic. *Journal for Research in Mathematics Education, 25*(2), 116-140.

Guin, D., & Trouche, L. (1999). The Complex Process of Converting Tools into Mathematical Instruments: The Case of Calculators. *International Journal of Computers for Mathematical Learning, 3*, 195-227.

Habre, S., & Abboud, M. (2006). Students' conceptual understanding of a function and its derivative in an experimental calculus course. *Journal of Mathematical Behavior, 25*, 57-72.

Harel, I. (1991). *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school.* Norwood, NJ: Ablex Publishing.

Harel, I., & Papert, S. (Eds.). (1991). *Constructionism*: Ablex Publishing.

Heid, M. K. (1984). *An Exploratory to examine the effects of resequencing skills and concepts in an applied calculus curriculum through the use of the microcomputer.* University of Maryland.

Heid, M. K. (1988). Resequencing skills and concepts in applied calculus using the computer as a tool. *Journal for Research in Mathematics Education, 19*(1), 3-25.

Heid, M. K., Blume, G. W., Hollebrands, K., & Piez, C. (2002). Computer Algebra Systems in Mathematics Instruction: Implications from Research. *Mathematics Teacher, 95*(8), 586-591.

Heid, M. K., & Edwards, M. T. (2001). Computer Algebra Systems: Revolution or Retrofit for Today's Mathematics Classrooms? *Theory into Practice, 40*(2), 128-.

Ioannidou, A., Repenning, A., Lewis, C., Cherry, G., & Rader, C. (2003). Making Constructionism Work in the Classroom. *International Journal of Computers for Mathematical Learning, 8*, 63-108.

Johnson, D., & Johnson, R. (2004). Cooperation and the use of Technology. In D. H. JONASSEN (Ed.), *Handbook of Research on Educational Communications and Technology* (pp. 785-811). Mahwah, NJ: Lawrence Erlbaum Associates.

Johnson, D. C. (2000). Algorithmics and programming in the school mathematics curriculum: support is waning - is there still a case to be made? *Education and Information Technologies, 5*(3), 201-214.

Jonassen, D., Carr, C., & Yueh, H.-P. (1998). Computers as Mindtools for Engaging Learners in Critical Thinking. *TechTrends, 43*(2), 24-32.

Jones, K. (2005). Graphing Calculators in the Teaching and Learning of Mathematics: a research bibliography. *Micro Math, 21*(2), 31-33.

Kennedy, D. (2002). AP Calculus and Technology: A Retrospective. *Mathematics Teacher*, *95*(8), 576-581.

Kurland, D. M., Pea, R. D., Clement, C., & Mawby, R. (1986). A study of the development of programming ability and thinking skills in high school students. *Journal of Educational Computing Research*, *2*, 429-458

Lawler, R. W. (1986). *Cognition and Computers: Studies in Learning*. New York: Halstead Press.

Lebesque, H. (1966). *Measure and Integral*. San Francisco: Holden-Day.

Leinbach, L. C., Hundhausen, J. R., Ostebee, A. M., Seneschal, L. J., & Small, D. (Eds.). (1991). *The Laboratory Approach to Calculus* (Vol. 20): Mathematical Association of America.

Lesh, R., Post, T., & Behr, M. (1987). Representations and Translations among Representations in Mathematics Learning and Problem Solving. In C. Janvier (Ed.), *Problems of Representations in the Teaching and Learning of Mathematics* (pp. 33-40). Hillsdale, NJ: Lawrence Erlbaum.

Li, L., & Tall, D. (1993). *Constructing Different Concept Images of Sequences and Limits by Programming.* Paper presented at the Proceedings of PME 17, Japan.

Liao, Y. K., & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Research on Computing in Education, 7*(3), 367-380.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic Inquiry*: Sage Publications, Inc.

Linn, M. C., & Dalbey, J. (1985). Cognitive consequences of Programming Instruction: Instruction, Access, and Ability. *Educational Psychologist*, *20*(4), P. 191, 116p, 195 charts, 192 graphs.

Maplesoft. (1988). Maple: Waterloo Maple Inc.

McCoy, L. P., & Dodl, N. R. (1989). Computer programming experience and mathematical problem solving. *Journal of Research on Computing in Education, 22*(1), 15-27.

Merriam, S. (1998). *Qualitative Research and Case Study Applications in Education*. San Francisco, CA: Jossey-Bass.

Miles, M. B., & Huberman, M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook* (2nd Edition ed.): Sage Publications, Inc; 2nd edition (1994).

Moersch, C. (2001). Next Steps: Using LoTI as a Research Tool. *Learning & Leading with Technology, 29*(3), 22-27.

Monaghan, J., Sun, S., & Tall, D. (1994). *Construction of the Limit Concept with a Computer Algebra System.* Paper presented at the Proceedings of PME 18, Lisbon.

Mor, Y., Hoyles, C., Kahn, K., Noss, R., & Simpson, G. (2004). Thinking in progress. [-20040701]. *Micro Math, 20*(2), 17-23.

NCTM. (2003). *Principles and Standards for School Mathematics* (3 ed.): National Council of Teachers of Mathematics.

Noss, R. (1997). Meaning mathematically with computers. In P. Bryant & Y. Nunes (Eds.), *Learning and Teaching Mathematics: An International Perspective* (pp. 289-314). Hove, UK: Psychology Press.

Noss, R., & Hoyles, C. (1992). Afterward: Looking back and looking-forward. In C. Hoyles & R. Noss (Eds.), *Learning Mathematics and Logo* (pp. 427-468). Cambridge, Ma.

Orton, A. (1983). Students' understanding of differentiation. *Educational Studies in Mathematics, 14*(3), 235-250.

Palmiter, J. R. (1986). *The impact of a computer algebra system on college calculus.* Unpublished Doctoral Dissertation, The Ohio State University.

Papert, S. (1980). Computer-based microworlds as incubators for powerful ideas. In R. Taylor (Ed.), *The computer in the school: Tutor, tool, tutee* (pp. 203-210). New York: Basic Books.

Park, K., & Travers, K. J. (1996). A comparative study of a computer-based and a standard college first-year calculus course. *CBMS Issues in Mathematics Education, 6*, 155-176.

Parnafes, O., & Disessa, A. (2004). Relations between Types of Reasoning and Computational Representations. *International Journal of Computers for Mathematical Learning, 9*(3), 251-280.

Pea, R. D. (1985). Beyond Amplification: Using the Computer to Reorganize Mental Functioning. *Educational Psychologist, 20*(4), 167-182.

Pea, R. D., & Kurland, D. M. (1984). On the Cognitive Effects of Learning Computer Programming. *New Ideas in Psychology, 2*(2), 137-168.

Pea, R. D., Kurland, D. M., & Hawkins, J. (1987). Logo and the Development of Thinking Skills. In R. D. Pea & K. Scheingold (Eds.), *Mirrors of Minds: Patterns*

of Excellence in Educational Computing (pp. 178-197). New Jersey: Ablex Publishing Corporation.

Perkins, D. N., & Salomon, G. (1992). Transfer Of Learning. from citeseer.ist.psu.edu/484081.html

Ploger, D. (1991). Learning About the Genetic Code Via Programming: Representating the Process of Translation. *Journal of Mathematical Behavior, 10*(1), 55-77.

Ploger, D., & Carlock, M. (1996). Mathematics and Molecules: Exploring Connections via Programming. *Journal of Mathematical Behavior, 15*, 441-452.

Rieber, L. P. (2003). Microworlds. In D. Jonassen (Ed.), *Handbook of Research for Educational Communications and Technology* (2 ed., pp. 583-603). Mahwah, NJ: Larwence Erlbaum Associates.

Salomon, G., & Perkins, D. (1989). Rocky Roads to Transfer: Rethinking Mechanisms of a Neglected Phenomenon. *Educational Psychologist, 24*(2), 113-142.

Salomon, G., Perkins, D., & Globerson, T. (1991). Partners in Cognition: Extending Human Intelligence with Intelligent Technologies. *Educational Researcher, 20*(3), 2-9.

Schoenfeld, A. H. (2002). Research Methods in (Mathematics) Education. In L. D. English (Ed.), *Handbook of International Research in Mathematics Education* (pp. 435-487). Mahwah: NJ: Lawrence Erlbaum Associates.

Schrock, C. S. (1989). *Calculus and computing: An exploratory study to examine the effectiveness of using a computer algebra system to develop increased conceptual understanding in a first-semester calculus course.* Unpublished Doctoral Dissertation, Kansas State University.

Sfard, A. (1991). On the Dual Nature of Mathematical Conceptions: Reflections on Processes and Objects as Different Sides of the Same Coin. *Educational Studies in Mathematics, 22,* 1-36.

Sfard, A. (1992). Operational origins of mathematical objects and the quandary of reification- the case of the function. In G. Harel & E. Dubinsky (Eds.), *The Concept of Function: Aspects of Epistemology and Pegagogy* (Vol. MAA Notes 25, pp. 59-84). Washington, DC: MAA.

Sfard, A., & Leron, U. (1996). Just give me a computer and i will move the earth: Programming as a catalyst of a cultural revolution in the mathematics classroom. *International Journal of Computers for Mathematical Learning, 1*(2), 189-195.

Sherin, B. L. (2001). A Comparison of Programming Languages and Algebraic Notation as Expressive Languages for Physics. *International Journal of Computers for Mathematical Learning, 6,* 1-61.

Simpson, G., Hoyles, C., & Noss, R. (2005). Designing a programming-based approach for modelling scientific phenomena. *21*, 143-158, 116p.

Soloway, E. (1993). Should we teach students to program? *Communications of the ACM, 36*(10), 21-24.

Stake, R. (1995). *The Art of Case Study Research*. Thousand Oaks, CA: Sage Publications.

Stein, M. K., Smith, M. S., Henningsen, M. A., & Silver, E. A. (2000). *Implementing Standards-based Mathematics Instruction: A casebook for Professional Development*. New York: Teachers College Press.

Sylvia, A. S. (1986). *Programming for learning in mathematics and science*. Paper presented at the Proceedings of the seventeenth SIGCSE technical symposium on Computer science education.

Tall, D., & Vinner, S. (1981). Concept Image and Concept Definition in Mathematics with particular reference to Limits and Continuity. *Educational Studies in Mathematics, 12*, 151-169.

Thomas, R. A., & Upah, S. C. (1996). Give programming instruction a chance. *Journal of Research on Computing in Education, 29*(1), 96, 13p, 95bw.

Topping, K., & Ehly, S. (Eds.). (1998). *Peer-Assisted Learning*. Mahwah, NJ: L. Erlbaum Associates.

Waits, B. K., Demana, F., & Kutzler, B. (1997). Guest Editorial. *The International Journal of Computer Algebra in Mathematics Education, 4*(1), 4-6.

Weller, M., Clark, J., Dubinsky, E., Loch, S., McDonald, M. A., & Merkovsky, R. (2000). An Examination of Student Performance Data in Recent RUMEC Studies. from http://www.math.kent.edu/~edd/Performance.pdf

Werner, L. L., Denner, J., & Bean, S. (2004). *Pair-programming Strategies for Middle School Girls.* Paper presented at the Computers and Advanced Technology in Education, Kauai, Hawaii.

Williams, L., & Kessler, R. R. (2003). *Pair Programming Illuminated*: Addison Wesley.

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In Support of Pair Programming in the Introductory Computer Science Course. *Journal of Computer Science Education*.

Williams, S. R. (1991). Models of limit held by college calculus students. *Journal for Research in Mathematics Education, 22*(3), 219-236.

Yin, R. (2002). *Case Study Research: Design and Methods* (3rd ed.): Sage Publications.

Zimmermann, W. (1991). Visual Thinking in Mathematics. In W. Zimmermann & S. Cunningham (Eds.), *Visualization in Teaching and Learning Mathematics* (pp. 127-137). Washington, D.C.: Mathematical Association of America.

APPENDIXES

APPENDIX A
Preliminary Questionnaire

Preliminary Written Questionnaire

1. (*Programming Experience*)
   a. What experience, if any, do you have programming a computer (this includes programming a calculator)? If none, please state so.
   b. If you have prior experience, please describe what uses of programming you have seen or utilized.

2. (*CAS experience*) Have you ever used a computer algebra system? YES or NO

3. (*Technology Uses in Math Education*)
   What technology have you used in your prior mathematics classes? Describe, in as much detain as you can, how that technology was used.

4. (*Technology's Role in Math Education*)
   Do you think the computer (or calculator) has an important role in learning mathematics? What do you think that role is?

5. (*Untapped Potential of Technology in Math Education*)
   Do you see other uses for technology in the mathematics classroom that have not been utilized in your prior classes?

6. (*Collaborative Work*)
   a. Have you worked collaboratively with other students (i.e. groups, pairs, etc.) in your prior math classes? In what way? Describe one situation.
   b. What do you feel are the benefits of working as a group?
   c. What do you feel are the problems with working in a group?

APPENDIX B
Maple labs

Maple Lab #1 (All Groups)

In this lab you will learn to define, evaluate, and plot functions in the Maple computer algebra system (CAS). You will work with your lab partner using the pair-programming strategy outlined by your instructor. You are to change roles approximately every 15 minutes.

I. Open a new Maple worksheet and address the following questions. Type the actual question text in the document so as to gain some practice working in Maple.
II. When you are finished, save the worksheet into your home folder (in the Maple folder you created) using the name "Lab 01 Group XX" where XX is the name of your group. For example, group B in 3rd period would save the file as "Lab 01 Group 3B"

Questions:

1. Consider the function $f(x) = x - 1$.
   a. Write a sequence of statements in Maple that implements this function.
   b. Have Maple evaluate the function at x = 3.5, 3.25, 3.1, 3.01, 3.001.
   c. Have Maple evaluate the function at x = 2.5, 2.75, 2.9, 2.99, 2.999.
   d. Produce a plot in a [-5,5] by [-5,5] window in blue.
   e. Explain how, **using the graph**, you evaluate f(4).
   f. Using the work you performed in steps (a-e), approximately what is $\lim_{x \to 3^+} f(x)$? Explain how you are estimating this; i.e. indicate specifically which of (a-e) you are basing your estimate on.

2. Consider the function $g(x) = 3x + 5$
   a. Write a sequence of statements in Maple that implements this function.
   b. Have Maple evaluate the function at x = 3.5, 3.25, 3.1, 3.01, 3.001.
   c. Have Maple evaluate the function at x = 2.5, 2.75, 2.9, 2.99, 2.999.
   d. Produce a plot in a [-5,5] by [-20,20] window in **black**.
   e. Explain how, **using the graph**, you evaluate f(4).

3. Consider the function $h(x) = \dfrac{1}{x-2}$.
   a. Write a sequence of statements in Maple that implements this function.
   b. Have Maple evaluate the function at x = 2.5, 2.25, 2.1, 2.01, 2.001.
   c. Have Maple evaluate the function at x = 1.5, 1.75, 1.9, 1.99, 1.999.

d. Estimate the following limits, $\lim_{x \to 2+} h(x)$ and $\lim_{x \to 2^-} h(x)$ and explain how you are estimating these limits.
e. Do you think $\lim_{x \to 2} h(x)$ exists? Why?
f. Produce a plot in a [-5,5] by [-5,5] window in blue.
g. Explain how the plot supports your answer to (d and e).

4. Consider the function $k(x) = g(f(x))$. (function f and g from #1,2)
   a. Write a sequence of statements in Maple that implement this function.
   b. Evaluate k(x) at x = 0, 1, 2, 3.
   c. Explain in words what you think happens (what is the sequence of events) when you ask Maple to evaluate the expression g(f(2)).
   d. Explain how, using ONLY the plots of f(x) and g(x), you could evaluate g(f(2)). Plot both graphs of f and g and explain how the two graphs could be used to evaluate g(f(2)).

5. Consider the function $l(x) = h(f(x))$ (function f and g from #1,3)
   a. Write a sequence of statements in Maple that implement this function.
   b. Have Maple evaluate the function at x = 2.5, 2.25, 2.1, 2.01, 2.001.
   c. Have Maple evaluate the function at x = 1.5, 1.75, 1.9, 1.99, 1.999.
   d. Estimate the following limits, $\lim_{x \to 2+} l(x)$ and $\lim_{x \to 2^-} l(x)$, and explain how you are estimating these limits.
   e. Do you think $\lim_{x \to 2} l(x)$ exists? Explain why?
   f. Make a plot of l(x) and explain how the graph is consistent with your conclusions from parts (c) and (d).
6. Consider the function $m(x) = \left(h(x)\right)^2$
   a. Write a sequence of statements in Maple that implement this function.
   b. Have Maple evaluate the function at x = 2.5, 2.25, 2.1, 2.01, 2.001.
   c. Have Maple evaluate the function at x = 1.5, 1.75, 1.9, 1.99, 1.999.
   d. Estimate the following limits, $\lim_{x \to 2+} m(x)$ and $\lim_{x \to 2^-} m(x)$, and explain how you are estimating these limits.
   e. Do you think that $\lim_{x \to 2} m(x)$ exists? Explain why?
   f. Make a plot of m(x) and explain how the plot is consistent with your conclusions from parts (c) and (d).
   g. Write a sequence of statements in Maple the implement this function m(x) using the previously defined function h(x). That is, your function definition should make direct reference to function h(x).
   h. Have Maple evaluate the function in part (g) at x =2.5 and 2.25.

Maple Lab #2 (P Groups)

[DO THIS FIRST] Import a library of Mysterious Functions

In this section, you will have Maple load a collection of mysterious functions which you are to explore. You will find that you are unable to plot the functions (TRY IT) or see symbolic definitions of the function.

```
> restart;
> libname := "S:/Student Work/Classes/deCastro/Maple/MysteryFunctions",
libname:
> with(MysteryFunctions):
```

Exploring a "*Mysterious*" function using evaluation.
In the section, you are to explore the behavior of function f(x). You are to consider the functions behavior at the points x = 1, 2, and -3. The only permitted action involving the function is evaluation, i.e. to evaluate a function f(x) at x=10;

```
> f(10);
```

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."
\>
\>
Description of behavior:

2) Explore and Describe the behavior of function f(x) at and near x=2.
\>
\>
Description of behavior:

3) Explore and Describe the behavior of function f(x) at and near x= -3.
\>
\>
Description of behavior:

Key Concepts – Procedures

Introduction to MAPLE Procedures
In last week's project, recall that you evaluated a function at several points close to a given number. Let's suppose that we want to evaluate the cosine function at several points nearer and nearer to x=0, specifically x= 0.1, 0.01, and 0.001. In the last lab we did the following.

```
> cos(0.1); cos( 0.01); cos( 0.001);
```

$$0.9950041653$$

$$0.9999500004$$

$$0.9999995000$$

This sequence of statements can be packaged so that it can be recalled at any time using Maple <u>procedures</u>. A procedure is essentially a function that packages together a sequence of instructions. Consider the following code segment.

```
> myCosineEval := proc()
   cos(0.1);
   cos(0.01);
   cos(0.001);
end proc:
```

To use this procedure, we must call it. That is we must instruct Maple to call the procedure into action. To call this procedure, you simply call the procedure as a function with NO inputs (we will call the inputs (if there are any) "parameters" to the procedure). Type `myCosineEval()` on the following line and have Maple execute the procedure. Notice that the parentheses are required.

```
>
>
>
```

Notice that all the results, were not printed. Which result was printed?

ANSWER:

When Maple performs (executes) the procedure, it returns the LAST value it calculates as the result. We can force Maple to print all the results using the print statement.

```
> myCosineEval := proc()
   print( cos(0.1) );
   cos(0.01);
   cos(0.001);
end proc:
> myCosineEval();
```

$$0.9950041653$$

$$0.9999995000$$

Notice that now we see the result that was printed as well as the final result computed. Try adding additional print statements to the procedure definition below.

```
> myCosineEval := proc()
   print( cos(0.1) );
   cos(0.01);
   cos(0.001);
end proc:
> myCosineEval();
```

$$0.9950041653$$

$$0.9999995000$$

To further extend this idea, suppose that we wanted to evaluate the cosine around other points beside 0. Suppose we would like to evaluate the function near the point x=5, specifically x = 5.1, 5.01, 5.001. Modify the procedure below to perform this computation. (You may also need to make a few other additions.)

```
> myCosineEval := proc()
   print( cos(0.1) );
   cos(0.01);
   cos(0.001);
end proc:
```

Now what if you wanted to evaluate the function near x=4, specifically x = 4.1, 4.01, 4.001. Modify the procedure below to perform this computation. (you may also need to make a few other additions.)

```
> myCosineEval := proc()
   print( cos(0.1) );
   cos(0.01);
   cos(0.001);
end proc:
```

Are you tired of typing yet? What if you wanted to perform the evaluation around many additional points? Clearly changing your procedure every time you want to use a different point is very tedious. Wouldn't it be nice if you could send the procedure information about the point you want to evaluate about and have it automatically adapt itself to that point?

Maple provides a simple way to address this issue. Let's add a parameter to our procedure. A parameter is input given to the procedure for it to use in performing its assigned job.
We add a parameter, named x, to the procedure as follows.

```
> myCosineEval := proc( x )
   print( cos(x+0.1) );
   print( cos(x+0.01) );
   print( cos(x+0.001) );
end proc:
```

Now try myCosineEval(0) and myCosineEval(5). Did you get the same results as before?
>

To further generalize, what if we wanted the procedure to evaluate some other function besides cosine? We can also make the function itself a parameter to the procedure. Try to make the procedure pointEval shown below work. It should evaluate any function at the given point plus 0.1.

Add a definition to procedure pointEval below so that the following three calls to pointEval will work.

```
> pointEval := proc( f, x )
      # What goes here?
end proc:
```

When properly defined, the following four statements should work.

```
> pointEval(sin,1);  # should evaluate sin at 1.01.
> pointEval(sin,2);  # should evaluate sin at 2.01.
> pointEval(tan,1);  # should evaluate tan at 1.01.
> pointEval(f, 1);  # should evaluate mystery function f at 1.01.
```

Programming Task - create the simpleLimitTable procedure

Create a procedure named simpleLimitTable that takes two parameters, a function, f, and a point, x . The procedure should display the function evaluated at points 0.1, 0.01, and 0.001 units above and below the specified point.

For example, the following use of this procedure should result in the given output.

```
> a := proc(x) x^2; end proc:
> simpleLimitTable(a,1);
        .900000        2.123450
        .990000        2.123450
        .999000        2.123450
       1.001000        2.123450
       1.010000        2.123450
       1.100000        2.123450
```

TIP: To format output you can use Maple's printf *procedure.*

```
> a := 2.12345; b := -2;
```

$$a := 2.12345$$

$b := -2$

To print variable a using 8 characters and 3 decimal places. Print b using 6 columns and 2 decimal places.

```
> printf("%8.3f%6.2f", a, b );
   2.123 -2.00
```

Use this to format the numbers you display.

```
> myCosineEval := proc()
   printf( "%8.4f", cos(0.1) );
end proc:

> myCosineEval();
   .9950
```

YOUR PROCEDURE GOES BELOW HERE.

```
> simpleLimitTable := proc( f, x )

  # What goes here?

end proc:
```

Exploration of Functions using LimitTable

The following three problems are to be explored using the simpleLimitTable procedure you just wrote. For each function, you will be given specific points at which to explore the function. Write an explanation of the behavior of the graph at the indicated points using calls to your procedure to justify your conclusions. Indicate the presence of holes, vertical asymptotes, and other relevant behavior.

Problem #1 - Exploration of g(x)

Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.

```
>
>
>
```

Problem #2 - Exploration of h(x)

Explain the behavior of the graph h(x) at x= 0, and x= 3. Use your procedure simpleLimitTable as well as specific evaluations of h(x) to support your conclusions.

```
>
>
>
```

Problem #3 - Exploration of k(x)

Explain the behavior of the graph k(x) at x= 0, and x= 2. Use your procedure simpleLimitTable as well as specific evaluations of k(x) to support your conclusions.

```
>
>
>
```

Challenge Problem

Consider the mystery function m(x) at the point x=2. In this problem, you are permitted to plot the function.

1. Plot the function using a [-5,5] by [-10, 10] window.

>
>

2. Using simpleLimitTable, estimate the limit as x approaches 2.

>
>

3. Modify your simpleLimitTable procedure so that it evaluates the function at points

within 0.1, 0.01, 0.001, and 0.0001 of the given point. Call the new procedure simpleLimitTable2

Define the new procedure below.

```
> simpleLimitTable2 := proc( f, x )
   # add code here
end proc:
>
```

Using simpleLimitTable2, evaluate function m(x) at points within 0.1, 0.01, 0.001, and 0.0001 of the given point x=2.

>

4. Do you notice anything when you compare the results of step 2 and step 3?

EXPLAIN HERE.

5.Modify your simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point. Call the new procedure simpleLimitTable3.

Define the procedure simpleLimitTable3 here.

```
> simpleLimitTable3 := proc( f, x )



end proc:
>
>
```

Using simpleLimitTable3, evaluate function m(x) at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point x=2.

>
>

6. Do you notice anything when you compare the results of steps 2, 3, and 4?

EXPLAIN HERE.

7. Look more closely at the graph around the point $x=2$ and explain the results found in steps 1-6. Do this by producing as many plots as necessary to effectively explain the results.

EXPLAIN HERE.

8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit?

EXPLAIN HERE.

Maple Lab #2 (N Groups)

[DO THIS FIRST] Import a library of Mysterious Functions

In this section, you will have Maple load a collection of mysterious functions you will explore.
You will find that you are unable to plot the functions (TRY IT) or see symbolic definitions of the function.

```
> restart;
> libname := "S:/Student Work/Classes/deCastro/Maple/CalcToolbox",
           "S:/Student Work/Classes/deCastro/Maple/MysteryFunctions",
libname:
> with(MysteryFunctions): with( CalcToolbox):
```

Exploring a "Mysterious" Function using evaluation.

In last week's project, recall that you evaluated a function at several points close to a given number. Let's suppose that we want to evaluate the cosine function at several points nearer and nearer to x=0, specifically x= 0.1, 0.01, and 0.001. In the last lab we did the following.

```
> cos(0.1); cos( 0.01); cos( 0.001);
                                0.9950041653
                                0.9999500004
                                0.9999995000
```

From this you attempted to determine the limit of the function as x approached 0 from the right.

In the lab, you will explore the behavior of an unknown function f(x). You are to consider the functions behavior at the points x = 1, 2, and -3.
The only permitted action involving the function is **evaluation**, i.e. to evaluate a function f(x) at x=10;

```
> f(10);
```

1) Explore and Describe the behavior of function f(x) at and near x=1 by evaluating the function at "appropriate points."

```
>
>
>
```

2) Explore and Describe the behavior of function f(x) at and near x=2.

```
>
>
>
```

3) Explore and Describe the behavior of function f(x) at and near x=-3.

```
>
>
```

Exploratory Tool - The simpleLimitTable Procedure

In this lab, you will use an exploratory tool called, simpleLimitPlot, that will perform many of the relevant evaluations for you.

To begin lets define a function with which to experiment and understand this tool.

```
> a := x->x^2+1;
                          2
              a := x -> x  + 1
```

The simpleLimitTable tool will produce a convenient table of function evaluations near a given point. For example,

```
> simpleLimitTable( a, 1, 0.01);
              simpleLimitTable(a, 1, 0.01)
```

Notice how this procedure produced two columns of output. The first column lists that x coordinate at which the function a(x) was evaluated. The second column gives the value of the function at that point. Specifically, a(0.9)=1.81 and a(1.1) =2.21.

In the above call to the procedure, we specified the function name first, a, followed by the point at which to explore the function, x=1, followed by a desired closeness to which we should approach the point of interest, 0.1. In this case, we want to evaluate the function at points as close as 0.1 to x=1.

Let's look at another application of the procedure to ensure that you understand the output it produces.

```
> simpleLimitTable( a, 1, 0.001);
```

Exploration of Functions using LimitTable

The following three functions are to be explored using the **simpleLimitTable** procedure described previously. For each function, you will be given specific points at which to explore the function. **Write an explanation of the behavior of the graph at the indicated points using calls to your procedure to justify your conclusions. Indicate the presence of holes, vertical asymptotes, and other relevant behavior**.

Problem #1 - Exploration of g(x)
Explain the behavior of the graph g(x) at x= -1, x= 1, and x= 4. Use your procedure simpleLimitTable as well as specific evaluations of g(x) to support your conclusions.

```
>
>
>
```

Problem #2 - Exploration of h(x)
Explain the behavior of the graph h(x) at x= 0, and x= 3. Use your procedure simpleLimitTable as well as specific evaluations of h(x) to support your conclusions.

```
>
>
>
```

Problem #3 - Exploration of k(x)
Explain the behavior of the graph k(x) at x= 0, and x= 2. Use your procedure simpleLimitTable as well as specific evaluations of k(x) to support your conclusions.

Challenge Problem
Consider the mystery function m(x) at the point x=2. In this problem, you are permitted to plot the function.
1. Plot the function using a [-5,5] by [-10, 10] window.

```
>
>
```

2. Using simpleLimitTable, estimate the limit as x approaches 2.

```
>
>
```

3. Use the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, and 0.0001 of the given point.

```
>
>
```

4. Do you notice anything when you compare the results of step 2 and step 3?
EXPLAIN HERE.

5. Using the simpleLimitTable procedure so that it evaluates the function at points within 0.1, 0.01, 0.001, 0.0001, and 0.00001 of the given point.

```
>
>
```

6. Do you notice anything when you compare the results of steps 2, 3, and 4?
EXPLAIN HERE.

7. Look more closely at the graph around the point x=2 and explain the results found in steps 1-6. Do this by producing as many plots as necessary to effectively explain the results.
EXPLAIN HERE.

```
>
>
```

8. What would be necessary to ensure that you have in fact found the correct limit? Describe what it would take for you to be convinced that you have found the correct limit?
EXPLAIN HERE.

Maple Lab #3 (P Groups)

In this lab you will create two tools, called `leftLimit` and `rightLimit`, for exploring the left-hand and right-hand limit concepts. This lab should be undertaken AFTER you complete the pre-lab activities.

Initially you are given the following procedure template:

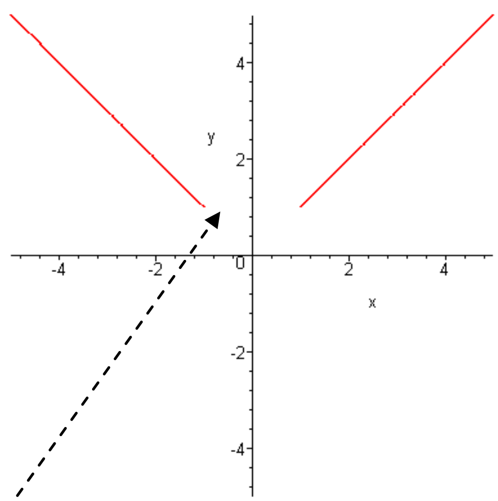
Mathematical Concept	Maple Exploratory Tool
$\lim_{x \to a^-} f(x)$	**leftLim := proc(f, a, n)** **YOU WILL COMPLETE THIS** **end proc:**
$\lim_{x \to a^+} f(x)$	**rightLim := proc(f, a, n)** **YOU WILL COMPLETE THIS** **end proc:**

Notice that, in the procedures, there will be 3 parameters, a function **`f`**, a point you are approaching, **`a`**, and the number, **`n`**. The procedure will produce a table of `x` and function values beginning at `a + 0.1` and ending at `a + 10^(-n)`.

Example 1:

Define the function

```
> v := proc( x )
    if type( x, realcons) then
       if x < -1 then
          -x;
       elif x > 1 then
          x;
       else
          undefined;
       end if:
    else
       'v(x)';
    end if:
  end proc:
```



$$v(x) = \begin{cases} -x, & x < -1 \\ x, & x > 1 \end{cases}$$

Explore the limits

```
> leftLim(v, -1, 3);
[Left Limit]   x              f(x)
--------------- ---------------
    -1.10000000       1.10000000
    -1.01000000       1.01000000
    -1.00100000       1.00100000
> rightLim(v,-1,3);
[Right Limit] x               f(x)
--------------- ---------------
     -.90000000              NaN
     -.99000000              NaN
     -.99900000              NaN
> leftLim(v,0,3);
[Left Limit]   x              f(x)
--------------- ---------------
     -.10000000              NaN
     -.01000000              NaN
```

NaN means "**N**ot **a** **N**umber" and implies that there is no point present.

```
-.00100000          NaN
```

Function Evaluation

```
> v1(1), v1(-2);
```
$undefined, 2$

PART I: Create the tools `leftLim` and `rightLim` to operate as shown above.

PART II: The library of mystery functions that you explored in the last lab will again be utilized in this lab. For each of the mystery functions, $f(x)$, $g(x)$, $h(x)$, $k(x)$, use the `leftLim` and `rightLim` tools to determine the behavior of the graph at each of the points requested below. You will find that you are unable to plot the functions; **the key idea in this part of the lab is to infer what you can from the left and right hand limits and using function evaluation.**

ON A SEPARATE SHEET OF PAPER, DRAW A SKETCH OF THE FUNTIONS BEHAVIOR NEAR THE REQUESTED POINTS (i.e. sketch the behavior of the function AT and AROUND the requested points; you are not sketching the entire graph!)

Function f(x)	**Function g(x)**	**Function h(x)**	**Function k(x)**
x= -3 x= 2	x= -1 x= +1 x= 4	x= 0 x= 3	x= 1 x=2

PART III: In this part of the lab, rather than exploring the provided mystery functions, you are charged with creating functions in Maple that have the desired behavior. You will then demonstrate that your functions have the desired behavior by giving appropriate supporting demonstrations using `leftLim`, `rightLim`, and appropriate plots. Your supporting arguments MUST USE ALL THREE methods.

Create Maple procedures implementing the following four functions. The functions must satisfy the given requirements at the given points.

Function b(x)	**Function c(x)**	**Function d(x)**	**Function e(x)**
Hole at x= 2 Vertical Asymptote at x=-1	Jump discontinuity at x= -1 $\lim_{x\to -3^-} c(x) = +\infty$ $\lim_{x\to -3^+} c(x) = +\infty$	Undefined on [1,2] $\lim_{x\to 1^-} d(x) = 2$ $\lim_{x\to 2^+} d(x) = 3$	$\lim_{x\to 1} e(x) = 3$ $e(1) = 5$ $\lim_{x\to -2^-} e(x) = +\infty$ $\lim_{x\to -2^+} e(x) = -\infty$

Your report should include:

1. The Maple code implementing `leftLim` and `rightLim` from part I.
2. Hand drawn sketches showing the behavior of the mystery functions around the given points from part II.
3. Maple output utilizing your (a) `leftLim` and `rightLim`, procedures and (b) relevant function evaluations that support and justify the sketches drawn in part II with discussion.
4. Maple definitions of the functions from part III with a written discussion of how you came up with functions `b(x), c(x), d(x), and e(x).`
5. Maple output utilizing (a) `leftLim`, (b) `rightLim`, (c) relevant function evaluations, and (d) plots of the functions that support and justify the function's behavior in part III.
6. Your answers to the following additional questions.

a. Are the Maple procedures `leftLim` and `rightLim` exact implementations of the corresponding mathematical limits? Explain.

b. When you use `leftLim` and `rightLim`, the procedures produce a sequence of x values that approach the requested x value. What behavior in the produced output indicates that a limit does or does not exist?

c. What does the value of the function `f(x)` at `x=a` have to do with the limit $\lim_{x \to a} f(x)$? Is the value the same, different, sometimes the same and sometimes different from the limiting value? Explain.

Maple Lab #3 (N groups)

In this lab you will use tools, called `leftLimit` and `rightLimit`, to explore the left-hand and right-hand limit concepts. This lab should be undertaken AFTER you complete the pre-lab activities.

Here are the two exploratory tools you will utilize along with the corresponding mathematical concept.

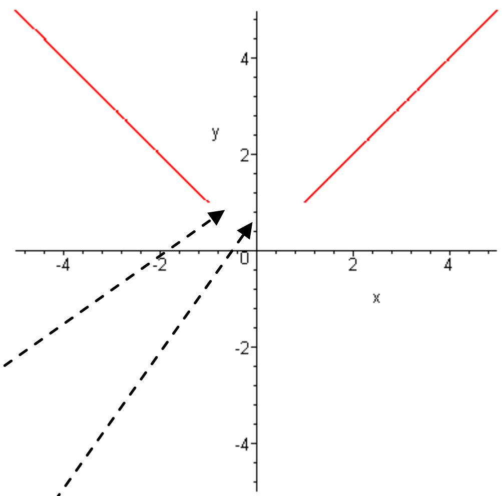
Mathematical Concept	Maple Exploratory Tool
$\lim\limits_{x\to a^-} f(x)$	**`leftLim( f, a, n)`**
$\lim\limits_{x\to a^+} f(x)$	**`rightLim(f, a, n)`**

Notice that in the procedures there are 3 parameters, a function **`f`**, a point you are approaching, **`a`**, and the number, **`n`**. The procedure will produce a table of x and function values beginning at `a + 0.1` and ending at `a + 10^(-n)`.

Example:

Define the function

```
> v := x->piecewise(x>1, x, x<-1,
-x, undefined);
```



Evaluation of limits

```
> leftLim(v, -1, 4);
[Left Limit]  x                f(x)
--------------- ---------------
    -1.10000000       1.10000000
    -1.01000000       1.01000000
    -1.00100000       1.00100000
    -1.00010000       1.00010000

> rightLim(v,-1,5);
[Right Limit] x                f(x)
--------------- ---------------
     -.90000000              NaN
     -.99000000              NaN
     -.99900000              NaN
     -.99990000              NaN
     -.99999000              NaN

> leftLim(v,0,5);
[Left Limit]  x                f(x)
--------------- ---------------
     -.10000000              NaN
     -.01000000              NaN
     -.00100000              NaN
     -.00010000              NaN
     -.00001000              NaN
```

NaN means "**N**ot **a** **N**umber" and implies that there is no point present.

Function Evaluation

```
> v(1); v(-2);
```

undefined

2

PART I: The library of mystery functions that you explored in the last lab will again be utilized in this lab. For each of the mystery functions, $f(x)$, $g(x)$, $h(x)$, $k(x)$, use the `leftLim` and `rightLim` tools to determine the behavior of the graph at each of the points requested below. You will find that you are unable to plot the functions; **the key idea in this part of the lab is to infer what you can from the left and right hand limits and by evaluating the function at key points.**

ON A SEPARATE SHEET OF PAPER, DRAW A SKETCH OF THE FUNTIONS BEHAVIOR NEAR THE REQUESTED POINTS (i.e. sketch the behavior of the function AT and AROUND the requested points; you are not sketching the entire graph!)

Function f(x)	**Function g(x)**	**Function h(x)**	**Function k(x)**
x= -3 x= 2	x= -1 x= +1 x= 4	x= 0 x= 3	x= 1 x=2

PART II: In this part of the lab, rather than exploring the provided mystery functions, you are charged with creating functions in Maple that have the desired behavior. You will then demonstrate that your functions have the desired behavior by giving appropriate supporting demonstrations using `leftLim`, `rightLim`, and appropriate plots. Your supporting arguments MUST USE ALL THREE methods.

Create Maple functions that have the stated limiting behavior. The functions must satisfy the given requirements at the specified points.

Function b(x)	**Function c(x)**	**Function d(x)**	**Function e(x)**
Hole at x= 2 Vertical Asymptote at x=-1	Jump discontinuity at x= -1 $\lim_{x\to -3^-} c(x) = +\infty$ $\lim_{x\to -3^+} c(x) = +\infty$	Undefined on [1,2] $\lim_{x\to 1^-} d(x) = 2$ $\lim_{x\to 2^+} d(x) = 3$	$\lim_{x\to 1} e(x) = 3$ $e(1) = 5$ $\lim_{x\to -2^-} e(x) = +\infty$ $\lim_{x\to -2^+} e(x) = -\infty$

Your report should include:

1. Hand drawn sketches showing the behavior of the mystery functions around the given points from part I.
2. Maple output utilizing the (a) `leftLim` and `rightLim`, procedures and (b) relevant function evaluations that support and justify the sketches drawn in part I with discussion.
3. Maple definitions of the functions from part II with a written discussion of how you came up with the functions.
4. Maple output utilizing (a) `leftLim`, (b) `rightLim`, (c) relevant function evaluations, and (d) plots of the functions that support and justify the behavior of functions `b(x), c(x), d(x), and e(x).`
5. Your answers to the following additional questions.
    a. Are the Maple procedures `leftLim` and `rightLim` exact implementations of the corresponding mathematical limits? Explain.
    b. When you use `leftLim` and `rightLim,` the procedures produce a sequence of x values that approach the requested x value. What behavior in the produced output indicates that a limit does or does not exist?
    c. What does the value of the function `f(x)` at `x=a` have to do with the limit $\lim_{x\to a} f(x)$? Is the value the same, different, sometimes the same and sometimes different from the limiting value? Explain.

Post-lab #3 questions (All groups)

The tables below were produced for a function $f(x)$. For each table, write a few sentences relating to the existence of the following three limits: $\lim_{x\to a^+} f(x)$, $\lim_{x\to a^-} f(x)$, and $\lim_{x\to a} f(x)$. Clearly indicate which limits exist, how you know (based upon what you see in the tables), and in situations where one or more of the limits fails to exist, describe what it is about the table the leads you to that conclusion. (e.g. if you believe that that limit as x approaches 10 is 20 then write $\lim_{x\to 10} f(x) = 20$ and explain how you conclude this fact using values in the respective table.) If no conclusion is possible then indicate so.

Table 1

x	f(x)
2.10000000	9.26100000
2.01000000	8.12060100
2.00100000	8.01200600
2.00010000	8.00120006
2.00001000	8.00012000
2.00000100	8.00001200
2.00000010	8.00000120
2.00000001	8.00000012

x	f(x)
1.90000000	6.85900000
1.99000000	7.88059900
1.99900000	7.98800600
1.99990000	7.99880006
1.99999000	7.99988000
1.99999900	7.99998800
1.99999990	7.99999880

Conclusions:

(a) if `f(2)=8` then …

(b) if `f(2)` is undefined then …

(b) if `f(2)=4` then …

Table 2

x	f(x)
2.10000000	4.87930340
2.01000000	4.08722195
2.00100000	4.00871339
2.00010000	4.00087125
2.00001000	4.00008712
2.00000100	4.00000871
2.00000010	4.00000087
2.00000001	4.00000009

x	f(x)
1.68000000	6.85900000
1.87800000	7.88059900
1.62000000	7.98800600
1.16200000	7.99880006
1.40500000	7.99988000
1.71600000	7.99998800
1.70100000	7.99999880
1.29000000	7.99999988

Conclusions:

Table 3

x	f(x)
2.10000000	4.87930340
2.01000000	4.08722195
2.00100000	4.00871339
2.00010000	4.00087125
2.00001000	4.00008712
2.00000100	4.00000871
2.00000010	4.00000087
2.00000001	4.00000009

x	f(x)
1.42100000	5.41468922
1.17400000	4.19665340
1.80400000	1.29502900
1.15200000	2.86328800
1.79700000	1.65979750
1.56900000	2.20207390
1.64100000	2.06293342
1.11000000	5.73533900

Conclusions:

Table 4

x	f(x)
2.10000000	16.11758758
2.01000000	12.50332216
2.00100000	10.07769600
2.00010000	17.57600000
2.00001000	8.21794983
2.00000100	16.71830269
2.00000010	13.16097188
2.00000001	9.48773561

x	f(x)
1.90000000	6.85900000
1.99000000	7.88059900
1.99900000	7.98800600
1.99990000	7.99880006
1.99999000	7.99988000
1.99999900	7.99998800
1.99999990	7.99999880
1.99999999	7.99999988

Conclusions:

Table 5

x	f(x)
2.10000000	5.41000000
2.01000000	5.04010000
2.00100000	5.00400100
2.00010000	5.00040001
2.00001000	5.00004000
2.00000100	5.00000400
2.00000010	5.00000040
2.00000001	5.00000004

x	f(x)
1.90000000	2.90000000
1.99000000	2.99000000
1.99900000	2.99900000
1.99990000	2.99990000
1.99999000	2.99999000
1.99999900	2.99999900
1.99999990	2.99999990
1.99999999	2.99999999

Conclusions:

Table 6

x	f(x)
2.10000000	7.94010000
2.01000000	5.49230400
2.00100000	5.88537600
2.00010000	7.47392100
2.00001000	7.74976400
2.00000100	5.79644900
2.00000010	5.32522500
2.00000001	3.48168900

x	f(x)
1.90000000	-2.72000000
1.99000000	-2.16200000
1.99900000	-1.09400000
1.99990000	-2.13800000
1.99999000	-2.28600000
1.99999900	-2.51000000
1.99999990	-2.68600000
1.99999999	-2.77000000

Conclusions:

Table 7		Table 8	
x	f(x)	x	f(x)
2.45700000	3.41000000	3.03300000	29.79100000
2.86300000	3.04010000	3.39700000	27.27090100
2.10900000	3.00400100	3.58900000	27.02700900
2.23300000	3.00040001	3.41400000	27.00270009
2.61300000	3.00004000	3.54600000	27.00027000
2.80000000	3.00000400	3.62800000	27.00002700
2.66000000	3.00000040	3.03500000	27.00000270
2.39600000	3.00000004	3.41900000	27.00000027
x	f(x)	x	f(x)
1.42100000	-2.80000000	2.75900000	24.38900000
1.24400000	-2.98000000	2.44800000	26.73089900
1.87600000	-2.99800000	2.62000000	26.97300900
1.83300000	-2.99980000	2.87000000	26.99730009
1.67800000	-2.99998000	2.93300000	26.99973000
1.86600000	-2.99999800	2.08900000	26.99997300
1.10700000	-2.99999980	2.41800000	26.99999730
1.25600000	-2.99999998	2.58100000	26.99999973
Conclusions:		Conclusions:	

Please answer the following questions.

A. Please mark **each** of the following six statements about limits as being **true** or **false**.

1. TRUE FALSE A limit describes how a function moves as x moves towards a certain point.
2. TRUE FALSE A limit is a number or point past which a function cannot go.
3. TRUE FALSE A limit is a number that the y-values of a function can be made arbitrarily close to by restricting x-values.
4. TRUE FALSE A limit is a number or point the function gets close to but never reaches.
5. TRUE FALSE A limit is an approximation that can be made as accurate as you wish.
6. TRUE FALSE A limit is determined by plugging in numbers closer and closer to a given number until the limit is reached.

B. Which of the above statements best describes a limit, as you understand it? (**Circle one**)

1 2 3 4 5 6 None

Maple Lab #4 (P Groups)

In this lab you will create a tool, called `dePlot`, for exploring the limit concept. This lab should be undertaken AFTER you complete the pre-lab activities.

Initially you are given the following procedure template. You will see this initial template in the file **Lab 04 P**. As you read through this document, type the indicated Maple statements in the section called creating the `dePlot` tool.

```
dePlot := proc( f, xLower, xUpper, L, e)
   # declare any necessary local variables here
   #
   local p1;
   # include graphics tools (display, and rectangle)
   #
   with( plottools ):
   with( plots ):

   # plot the graph
   #
   p1 := plot( f(x), x=xLower..xUpper, discont=true, thickness=2);

   # display the graph
   display([p1]);
end proc:
```

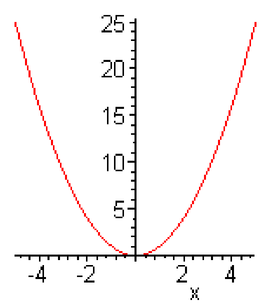
Notice that in the procedure there are 5 parameters, a function **`f`**, a lower and upper limit for the x range of the plot, **`xLower`**, **`xUpper`**, a limit value **`L`**, and a tolerance value named **`e`**. Here is the initial output of the procedure for the squaring function on the interval [-5,5].

Define the squaring function.

```
> f := proc(x)
   x^2;
end proc:
```

Have the `dePlot` tool produce the graph.

```
> dePlot(f, -5, 5, 4, 1);
Warning, the name arrow has been redefined
Warning, the name arrow has been redefined
```



Step 1: Modify `dePlot` to draw a horizontal line, y=L.

To accomplish this we will add an additional variable, `L1`, that will represent the line.

If you need additional information about the line function, simply issue maple the following command.

```
> ?line
```

plottools[line] - generate 2-D or 3-D plot object for a line segment

Calling Sequence

line(**a**, **b**, **options**)

Parameters

`a,b` - end points of the line segment

Description

- The routine **line** creates a plot data object which when displayed, is an line segment starting at point **a** and ending at point **b**. The line is two- or three-dimensional depending on whether **a** is a list of two or three values, respectively.
- A call to **line** produces a plot data object, which can be used in a **PLOT** or **PLOT3D** data structure, or displayed using the function **plots[display]**.
- Remaining arguments are interpreted as options which are specified as equations of the form **option = value**. These options are the same as for those found with the **plot** or **plot3d** command. See **?plot,options** and **?plot3d,options** for more information.
- The command with(plottools,line) allows the use of the abbreviated form of this command.

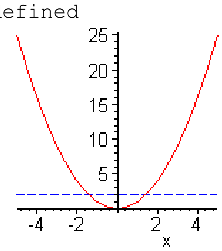
Examples

```
> with(plottools):
l := line([0,0], [3,4], color=red, linestyle=3);
plots[display](l);
```

See Also

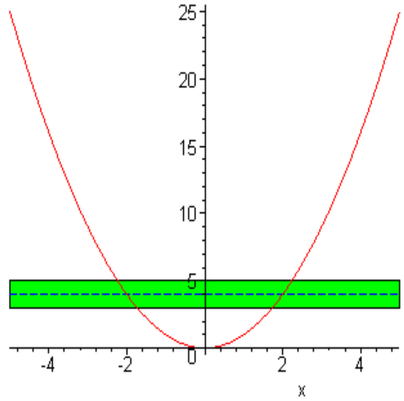
plottools, plot3d[structure], plot[structure], plots[display]

```
> dePlot := proc( f, xLower, xUpper, L, e)
    # declare any necessary local variables here
    #
    local p1, L1;

    # include graphics tools (display, and rectangle)
    #
    with( plottools ):
    with( plots ):

    # plot the graph
    #
    p1 := plot( f(x), x=xLower..xUpper, discont=true, thickness=2);

    # add the line y=L
    #
    L1 := line( [ xLower, L], [ xUpper, L], color=blue, linestyle=DASH);

    # display the graph and the line
    #
   display( [ p1, L1] );
end proc:

> dePlot(f, -5, 5, 4, 1);
Warning, the name arrow has been redefined
```



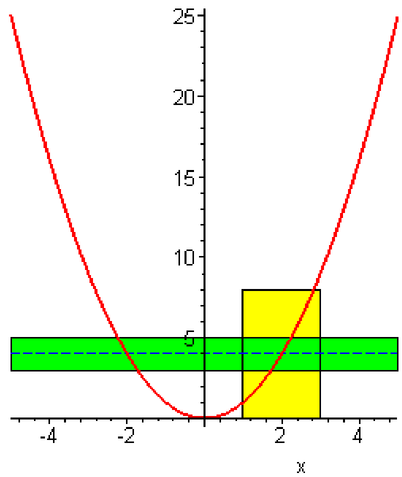
Notice the underlined statements.

Step 2: Look-up the `rectangle` command in Maple and modify the procedure to draw a rectangle that is centered on the line y=L that extends **`e`** units above and below **`L`**.



Thus, we now have `dePlot` shading the portion on the graph that is within **`e`** units of **`L`**.

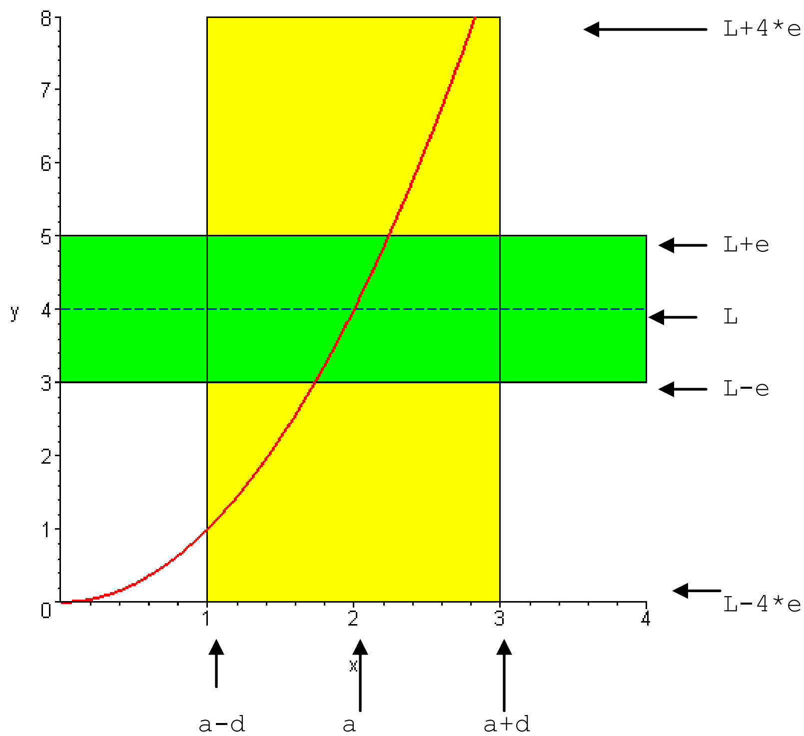
Step 3: Modify `dePlot` to take 2 additional parameters, **`a`** and **`d`**. Draw a rectangle centered on x=**`a`** extending horizontally **`d`** units around **`a`** and vertically from the x-axis to the y value that is **`4*e`** above L.



Now `dePlot` shades the portion on the graph that is *BOTH* within **`e`**=`1` units of **`L`**=`4` and the part within **`d`**=1 units of **`a`**=2.

Step 4: Since we will be focusing on the two shaded regions, modify `dePlot` so that the plot window is: **x** from `a-2*d` to `a+2*d` and the y-window is from `L-4*e`, `L+4*e`. This will provide sufficient space to observe the shaded regions. In doing so, the parameters `xLower` and `xUpper` will no longer be necessary so remove them! Thus the procedure will now have the following parameters:

```
> dePlot := proc( f, L, e, a, d)
```

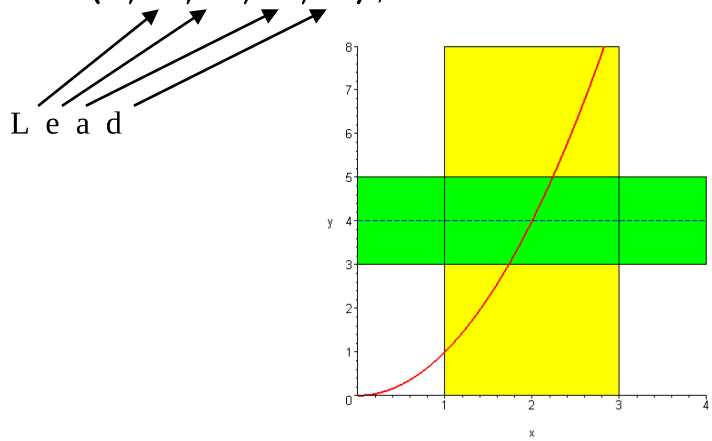


A Friendly Limit Argument

Suppose we wish to explore the limit $\lim_{x\to 2} x^2$. Using function $f(x)=x^2$, make a guess as to what the limit as x approaches **a**=2 is using the graph. Call your guess **L**. (Obviously you selected **L**=4.)

Now have `dePlot` draw the graph and shade the portion of the graph that is within **e**=1 unit of your tentative limit, **L**=4, and within **d**=1 units of **a**=2. Here is what we get.

```
> dePlot(f, 4, 1, 2, 1);
```



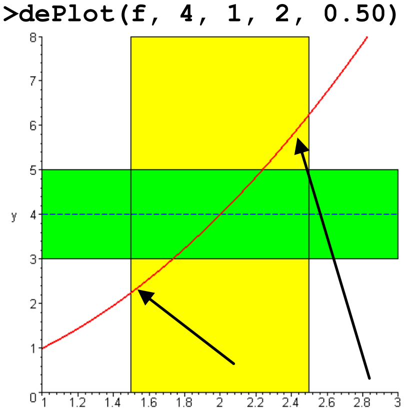
Does any part of the graph within the vertical rectangle fall outside the horizontal rectangle? YES! This means that if we let the x-values stray as much as **d**=1 unit from **a**=2, we are unable to keep the y-values to within **e**=1 of the proposed limiting value.

QUESTION: Is it possible to keep the x values near 2 and have all the corresponding y-values lie within **e**=1 of the limiting value? To do this, try varying ONLY the **d** value and see if it possible to find a **d** value for which all the y-values are within **e**=1 of the proposed limiting value.

After some experimentation with different values of **d**, we find…

Try d=0.50:

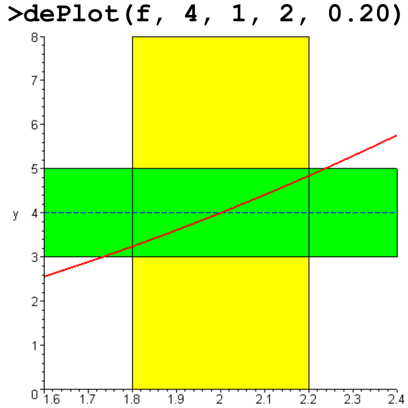
>dePlot(f, 4, 1, 2, 0.50); NOPE!



OOPS! Some of the graph lies outside the **e**-band

Try d=0.20:

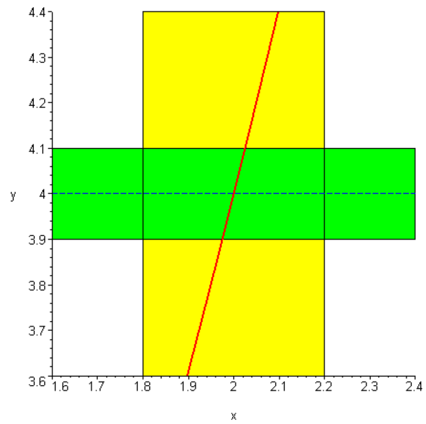
>dePlot(f, 4, 1, 2, 0.20); YES!



Good! None of the graph falls outside the **e**-band.

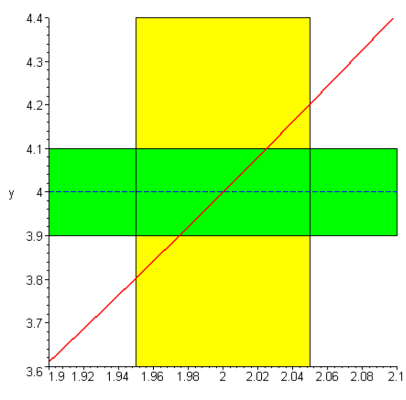
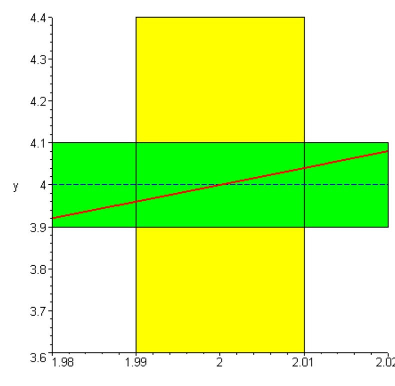
Thus **if we keep the x-values within `d`=0.20 of `a`=2 we are certain that the y-values are within `e`=1 unit of `L`=4**.

Next, lets see if it is possible to get even closer to the proposed limiting value `L=4` by keeping the x-values near `a=2`.
So now lets tighten our requirement of `e=1` and require that y-values be kept to within `e=0.10` of the proposed limiting value. Let's keep all the same parameter values except let's decrease `e` to 0.10. Now here's what we get.



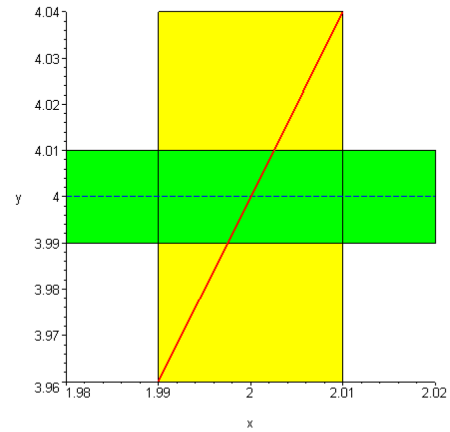
Clearly, it is not possible to keep the y-values within 0.10 of 4 without tightening our requirement on x. Let's see if we can find a suitable constraint on the x-values by countering this `e`-selection (e=0.1) with a new `d`-selection.

After some more experimentation with different `d` values, we find…

Try d=0.05:	**Try d=0.01:**
`>dePlot(f, 4, 1, 2, 0.05);` **`NOPE!`**	**`>dePlot(f, 4, 1, 2, 0.01);`** **`YES!`**
	

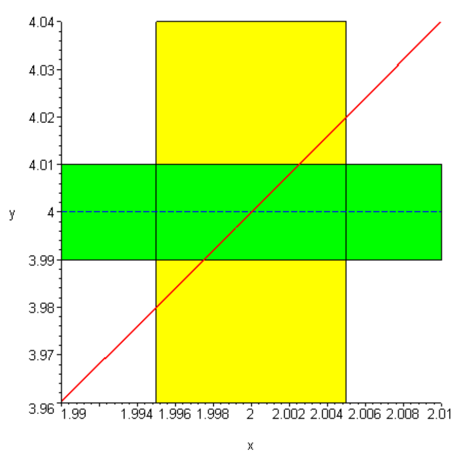
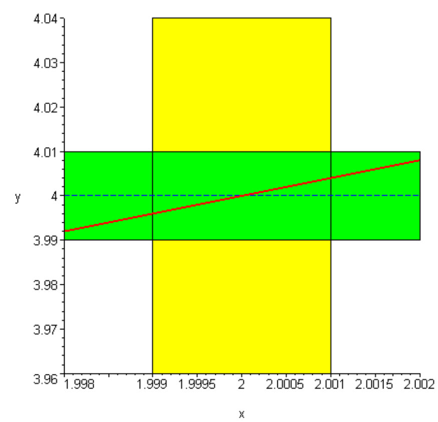
Thus it **is possible constrain the y-values to within `e=0.10` of `L=4` by requiring that the x-values be within 0.01 units of `a=2`**.

Continuing in the same manner, see if it is possible to get even closer to the proposed limiting value **L**=4 by keeping the x-values near **a**=2. So now lets tighten our requirement of **e**=1 and require that y-values be kept to within **e**=0.01 of the proposed limiting value. Keeping all the same parameter values decreasing e to 0.01, we see…



Clearly, it is not possible to keep the y-values within 0.10 of 4 without tightening our requirement on the x-values. Let's see if we can find a suitable constraint on the x-values by countering this **e**-selection with a new **d**-selection.

Experimenting with various values of **d**, we find…

<table>
<tr><th>Try d=0.005:</th><th>Try d=0.001:</th></tr>
<tr><td>>dePlot(f, 4, 0.01, 2, 0.005);
NOPE!

</td><td>>dePlot(f, 4, 0.01, 2, 0.001);
YES!

</td></tr>
</table>

Thus **it is possible constrain the y-values to within e=0.01 of L=4 by requiring that the x-values be within 0.001 units of a=2.**

Think about how long this argument, involving the repeated section of an **e**-value followed by a countermove involving a suitable **d**-value selection, must continue? What must happen in order for you to conclude that $\lim_{x \to a} f(x) = L$?

QUESTIONS AND ACTIVITIES

1. Consider the "friendly" argument just discussed, discuss how long must the argument involving the repeated section of an **e**-value followed by a countermove involving a suitable **d**-value selection continue? What must happen in order for you to conclude that $\lim_{x \to a} f(x) = L$?

Use the strategy outlined above to argue the following limits are correct or incorrect, respectively.

2. Create an argument using Maple that demonstrates that $\lim_{x \to 4} \sqrt{x} = 2$. Your argument show consist of a sequence of at least 4 `dePlot`s and explanations that support your claims. Start with **e**=0.5, **d**=5.

3. Argue that $\lim_{x \to 4} \sqrt{x} \neq 1.99$. Your argument show consist of a sequence of at least 4 `dePlot`s and explanations that support your claims. Start with **e**=0.5, **d**=5.
4. Create an argument using Maple that demonstrates that $\lim_{x \to 2} x^2 \neq 3$. Your argument show consist of a sequence of at least 4 `dePlot`s and explanations that support your claims. Start with **e**=1.
5. Determine the limit of $f(x) = \dfrac{|x-1|}{x-1}$ as x approaches 1 or indicate that the limit does not exist. In either case, provide a sequence of `dePlot`s and corresponding discussion relating to the existence of the limit. Clearly explain your conclusion in terms of the plots you produced.

6. For each of the following limits, find the largest **d** that ensures that the y-values of the function are within the specified **e** value of the limiting value. Find this by trial and error, producing a sequence of `dePlot`s that lead to and support your conclusion. **In each problem, begin with d=1.**

   a. $\lim_{x \to 4} 2x - 1 = 7$ for **e**=0.5 and for **e**=0.05

   b. $\lim_{x \to 4} \dfrac{x^2 - 16}{x - 4} = 8$ for **e**=0.5 and for **e**=0.05

   c. $\lim_{x \to 0} \left[x \cdot \sin\left(\dfrac{1}{x}\right) \right]$ for **e**=0.05
7. For limit problems in part 6a and 6b, find the **exact** largest **d**-value by using Maple's `solve` procedure. That is, have Maple solve the equation $|f(x) - L| < e$ for x (i.e. `solve(abs(f(x)-L)<e, x)`) and using the range of x-values returned, determine the exact largest value of **d** required to keep the function values within **e** units of **L**. Show that the answers you get are consistent with the ones you found by trial and error in part (6). Clearly explain how you determined the exact largest **d**-value.

8. In part 5, you found that the two-sided limit did not exist. Think about how you could modify the `dePlot` procedure to permit exploration of a one-sided limit.
   a. Create a new procedure called `dePlotLeft` that will permit you to argue for the existence of a **left-hand limit**. Do this by modifying the `dePlot` procedure in an appropriate way.

Maple Lab #4 (N Groups)

In this lab you will use a tool, called `dePlot`, for exploring the limit concept. This lab should be undertaken AFTER you complete the pre-lab activities.

```
> dePlot( f, L, e, a, d );
```

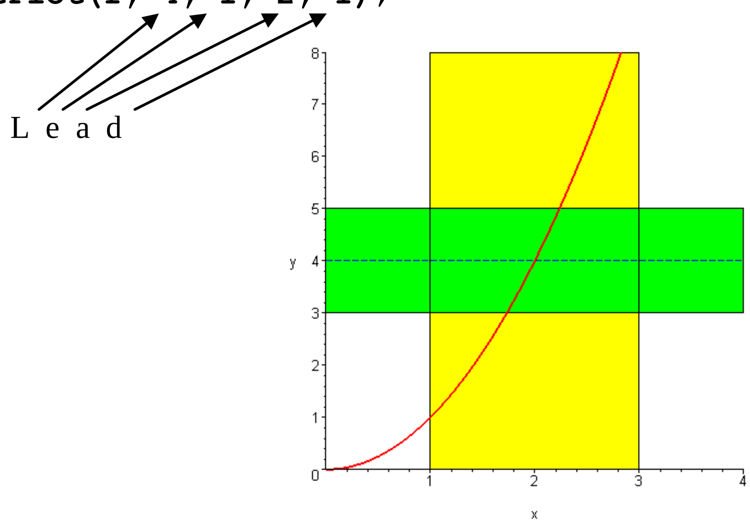
The dePlot procedure will provide a visual picture of the limiting process. To understand the visualization, consider the following debate.

A Friendly Limit Argument

Suppose we wish to explore the limit $\lim_{x \to 2} x^2$. Using function $f(x) = x^2$, make a guess as to what the limit as x approaches `a=2` is using the graph. Call your guess **L**. (Obviously you selected **L**=4.)

Now have `dePlot` draw the graph and shade the portion of the graph that is within **e**=1 unit of your tentative limit, **L**=4, and within **d**=1 units of **a**=2. Here is what we get.

```
> dePlot(f, 4, 1, 2, 1);
```



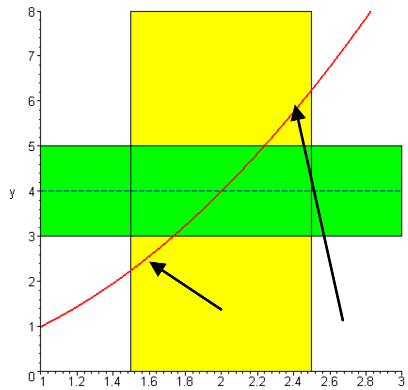
Does any part of the graph within the vertical rectangle fall outside the horizontal rectangle? YES! This means that if we let the x-values stray as much as **d**=1 unit from **a**=2, we are unable to keep the y-values to within **e**=1 of the proposed limiting value.

QUESTION: Is it possible to keep the `x` values near 2 and have all the corresponding y-values lie within **e**=1 of the limiting value? To do this, try varying ONLY the **d** value and see if it possible to find a **d** value for which all the y-values are within **e**=1 of the proposed limiting value.

After some experimentation with different values of **d**, we find…

Try d=0.50:

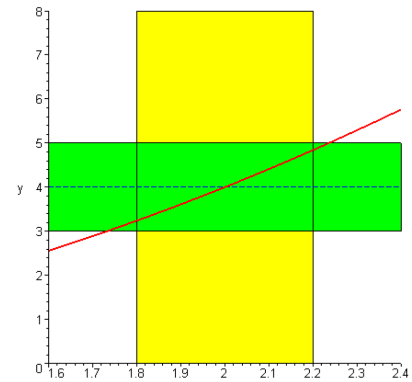
>dePlot(f, 4, 1, 2, 0.50);
NOPE!



OOPS! Some of the graph lies outside the **e**-band

Try d=0.20:

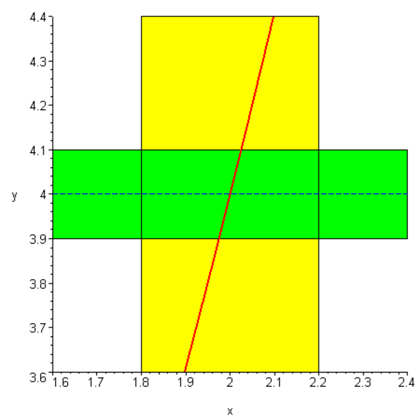
>dePlot(f, 4, 1, 2, 0.20);
YES!



Good! None of the graph falls outside the **e**-band.

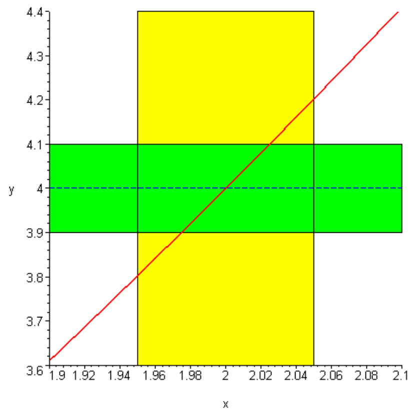
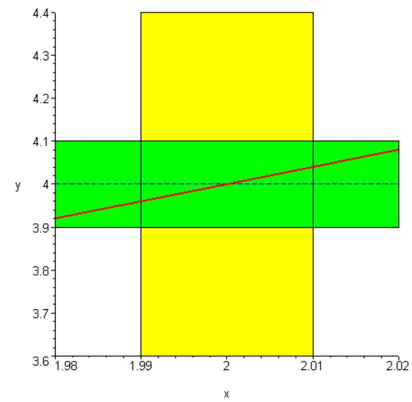
Thus **if we keep the x-values within d=0.20 of a=2 we are certain that the y-values are within e=1 unit of L=4**.

Next, let's see if it is possible to get even closer to the proposed limiting value **L**=4 by keeping the x-values near **a**=2.
So now lets tighten our requirement of **e**=1 and require that y-values be kept to within **e**=0.10 of the proposed limiting value. Let's keep all the same parameter values except let's decrease **e** to 0.10. Now here's what we get.



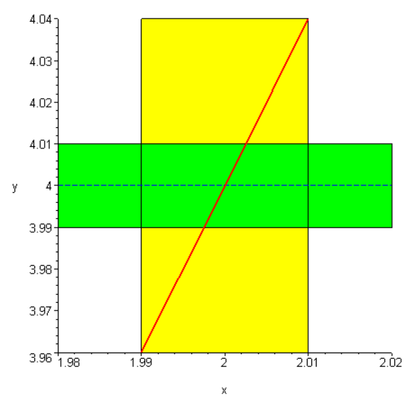
Clearly, it is not possible to keep the y-values within 0.10 of 4 without tightening our requirement on x. Let's see if we can find a suitable constraint on the x-values by countering this **e**-selection (e=0.1) with a new **d**-selection.

After some more experimentation with different **d** values, we find…

Try d=0.05:	**Try d=0.01:**
>dePlot(f, 4, 1, 2, 0.05); **NOPE!**	**>dePlot(f, 4, 1, 2, 0.01);** **YES!**
	

Thus it **is possible constrain the y-values to within e=0.10 of L=4 by requiring that the x-values be within 0.01 units of a=2**.

Continuing in the same manner, see if it is possible to get even closer to the proposed limiting value **L**=4 by keeping the x-values near **a**=2. So now lets tighten our requirement of **e**=1 and require that y-values be kept to within **e**=0.01 of the proposed limiting value. Keeping all the same parameter values decreasing e to 0.01, we see…

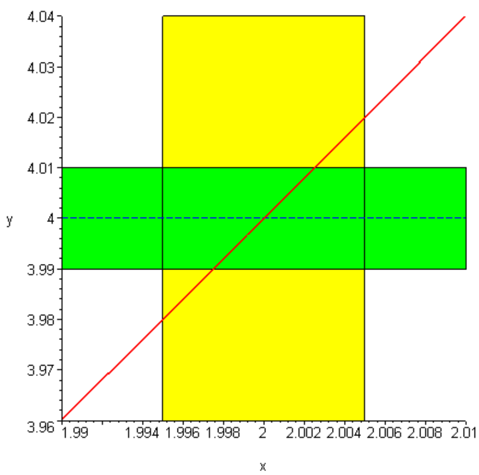


Clearly, it is not possible to keep the y-values within 0.10 of 4 without tightening our requirement on the x-values. Let's see if we can find a suitable constraint on the x-values by countering this **e**-selection with a new **d**-selection.

Experimenting with various values of **`d`**, we find…

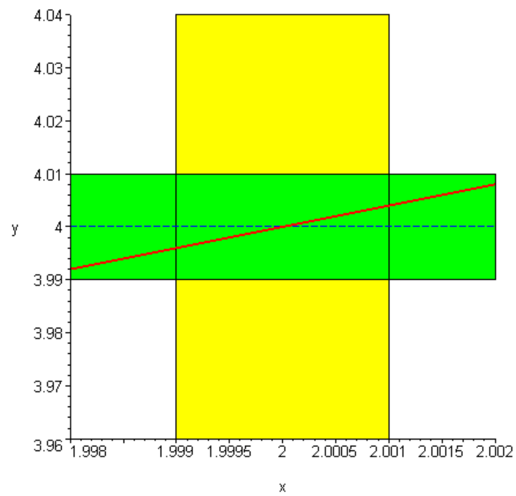
Try d=0.005:

```
>dePlot(f, 4, 0.01, 2, 0.005);
NOPE!
```



Try d=0.001:

```
>dePlot(f, 4, 0.01, 2, 0.001);
YES!
```



Thus **it is possible constrain the y-values to within `e=0.01` of `L=4` by requiring that the x-values be within 0.001 units of `a=2`.**

Think about how long this argument, involving the repeated section of an **e**-value followed by a countermove involving a suitable **`d`**-value selection, must continue? What must happen in order for you to conclude that $\lim_{x \to a} f(x) = L$?

QUESTIONS AND ACTIVITIES

9. Consider the "friendly" argument just discussed, discuss how long must the argument involving the repeated section of an **e**-value followed by a countermove involving a suitable **d**-value selection continue? What must happen in order for you to conclude that $\lim_{x \to a} f(x) = L$?

Use the strategy outlined above to argue the following limits are correct or incorrect, respectively.

10. Create an argument using Maple that demonstrates that $\lim_{x \to 4} \sqrt{x} = 2$. Your argument show consist of a sequence of at least 4 `dePlot`s and explanations that support your claims. Start with **e**=0.5, **d**=5.

11. Argue that $\lim_{x \to 4} \sqrt{x} \neq 1.99$. Your argument show consist of a sequence of at least 4 `dePlot`s and explanations that support your claims. Start with **e**=0.5, **d**=5.
12. Create an argument using Maple that demonstrates that $\lim_{x \to 2} x^2 \neq 3$. Your argument show consist of a sequence of at least 4 `dePlot`s and explanations that support your claims. Start with **e**=1.
13. Determine the limit of $f(x) = \dfrac{|x-1|}{x-1}$ as x approaches 1 or indicate that the limit does not exist. In either case, provide a sequence of `dePlot`s and corresponding discussion relating to the existence of the limit. Clearly explain your conclusion in terms of the plots you produced.

14. For each of the following limits, find the largest **d** that ensures that the y-values of the function are within the specified **e** value of the limiting value. Find this by trial and error, producing a sequence of `dePlot`s that lead to and support your conclusion. **In each problem, begin with d=1.**

    d. $\lim_{x \to 4} 2x - 1 = 7$ for **e**=0.5 and for **e**=0.05

    e. $\lim_{x \to 4} \dfrac{x^2 - 16}{x - 4} = 8$ for **e**=0.5 and for **e**=0.05

    f. $\lim_{x \to 0} \left[x \cdot \sin\left(\dfrac{1}{x}\right) \right]$ for **e**=0.05

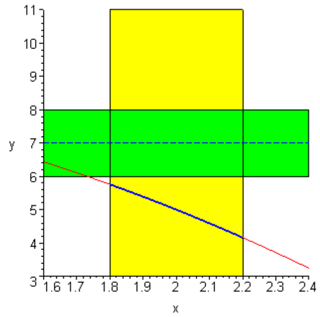
15. For limit problems in part 6a and 6b, find the **exact** largest **d**-value by using Maple's `solve` procedure. That is, have Maple solve the equation $|f(x) - L| < e$ for x (i.e. `solve(abs(f(x)-L)<e, x)`) and using the range of x-values returned, determine the exact largest value of **d** required to keep the function values within **e** units of **L**. Show that the answers you get are consistent with the ones you found by trial and error in part (6). Clearly explain how you determined the exact largest **d**-value.

Post-lab #4 Questions (All groups)

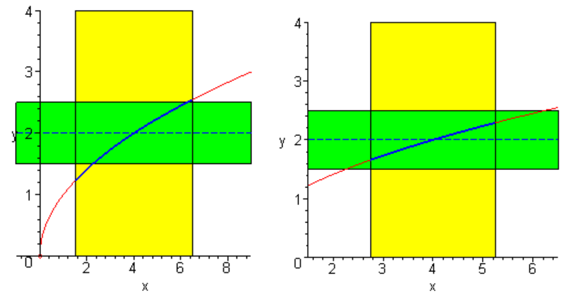
Due: on exam day.

1. Consider the following `dePlot`s. Indicate what the plot(s) tells you in each case (a)-(d).

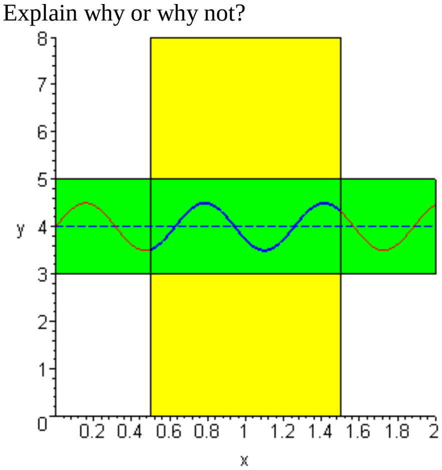
(a) The following `dePlot` says something about the limit of a function f(x) as x approaches some specific value. What is the value and what, if anything, does this plot tell you about the limit? Be as specific as you can.



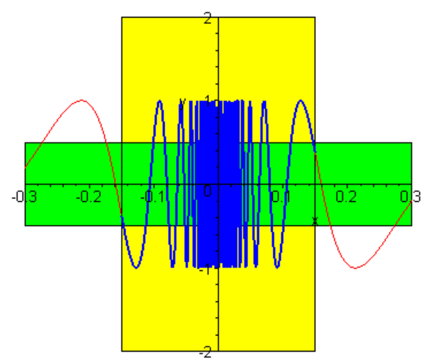
(b) What do the following two `dePlot`s suggest about a limit relating to function h(x)? Explain.



(c) Does this prove that $\lim_{x\to 1}\left(g(x)\right) = 4$? Explain why or why not?



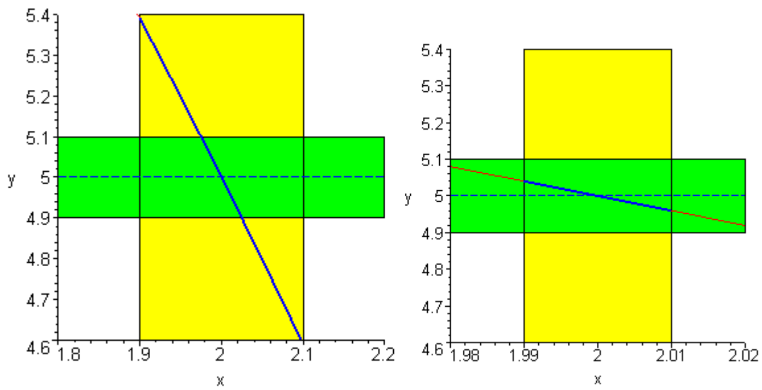
(d) The following `dePlot` says something about the limit of a function k(x) as x approaches some specific value. What is the value and what, if anything, does this plot tell you about the limit? Be as specific as you can.



2. For the function $f(x) = 9 - x^2$, explain how a `dePlot` relates to `leftLim` and `rightLim` tables? What, if anything, do the **rectangles in the `dePlots`** have to do with the **columns of the tables**? Discuss these using the tables and plots below.

[Left Limit] x	f(x)	[Right Limit] x	f(x)
1.90000000	5.39000000	2.10000000	4.59000000
1.99000000	5.03990000	2.01000000	4.95990000
1.99900000	5.00399900	2.00100000	4.99599900
1.99990000	5.00039999	2.00010000	4.99959999
1.99999000	5.00004000	2.00001000	4.99996000

dePlots



3. Here is the formal mathematical definition of limit.

DEFINITION (Limit): Let f be a function of x. We say

$$\lim_{x \to a} f(x) = L$$ (read "*the limit of f as x approaches a is L*")

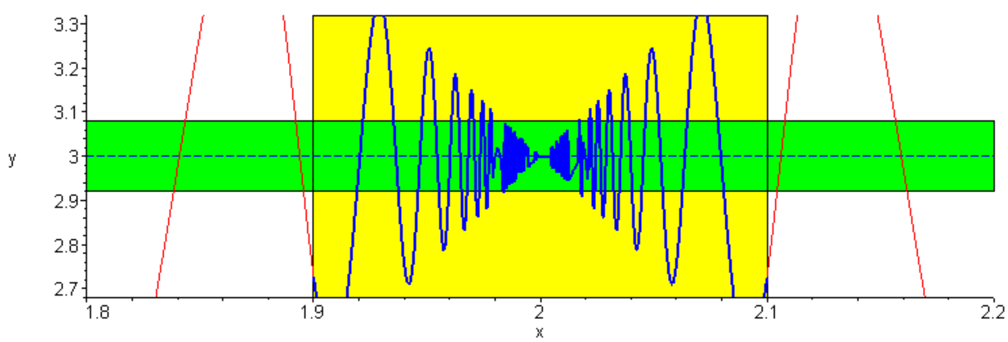
IF whenever we are given a positive real number **`e`**, we are able to find another positive number **`d`** so that |f(x) – L| < **`e`** for any x satisfying |x – a| < **`d`**.

We can make **`f(x)`** as close to **`L`** as we want, say **`e`**

$$\lim_{x \to a} f(x) = L$$

By keeping **`x`** get close enough to **`a`**, say **`d`**.

Explain how this definition relates to `dePlots` you created in the lab. Be as specific as you can in explaining how the rectangles in the `dePlots` relate to specific phrases in the limit definition. Use the sample `dePlot` for function `r(x)` below in your discussion.



Specifically, **how do the following phrases in the definition relate to rectangles in dePlots?** Discuss how L, e, a, and d enter into the `dePlots`.

Phrase 1: "whenever we are given a positive real number **`e`**"
Phrase 2: "we are able to find another positive number **`d`** so that |f(x) – L| < **`e`**"
Phrase 3: "for any x satisfying |x – a| < **`d`**"

APPENDIX C
Peer Evaluation Form

Lab: ________________________ Pair names: _____________________________________

Date: ________________

Partner being evaluated: ___________________________

For each question, please evaluate your partner over the course of the lab. Answer **each** question using the following 20 point scale:
0 (poor performance) to 20 (superior performance).

Score (0-20)	
	Your partner came to the lab having adequately prepared by reading preliminary materials prior to starting the lab
	Your partner cooperatively followed the pair-programming mode (rotating roles of driver and navigator).
	Your partner did their fair share of the work.
	Your partner made contributions towards completion of the lab assignment.
	You partner cooperated.
	TOTAL (x / 100)

Questions suggested in (L. Williams, et al., 2002)

Please feel free to add any comments related to your collaboration. What works well? What does not work well?

APPENDIX D
Source Code for Maple Procedures

Maple Source Code for Mystery Functions, f, g, h, k, and m.

A library of functions for students to explore VIA EVALUATION ONLY.

```
> MysteryFunctions := module()
   option package;
   export constant, f, g, h, k, m;

   # This function approximates a constant function f(x)=c using a
   # linear approximation that possesses a slope of ~0 (10^-16). It is
   # utilized to address a bug in Maple 8's limit function.  For some
   # reason, Maple is unable to evaluate limits of constant functions,
   # i.e. evalf(Limit( 1, x=0 )); will fail
   #
   constant := (c,x) -> c + 10.0^(-15)*x;

   # these functions are defined as a procedure so that students will
   # be unable to use the limit and plot functions. They must
   # explore the behavior of the function using evaluation only.

   f := proc(x)               # A function with hole at x=2, va at x=-3
      option Copyright;       # HIDE PROCEDURE CONTENTS!
      local y;

      if x > 2 or x < 2 then  # SNEAKY TRICK- to prohibit symbolic
         y := 10/(x+3);       # examination of function
         evalf(y);
      else
         undefined;
      fi:
   end proc:

   g := proc(x)
      option Copyright;          # HIDE PROCEDURE CONTENTS!
      if x <= -3 then
         x+5;
      elif x <=-1 then
         2;
      elif x > 1 and x<=4 then
         (x-1)^2+2;
      elif x > 4 then
         3;
      else
         undefined;
      fi:
   end proc:
```

```
  h := proc(x)
     option Copyright;          # HIDE PROCEDURE CONTENTS!
     if x <= 0 then
        x^2;
     elif x <=3 then
        2*x;
     elif x > 3 then
        1/(x-3);
     else
        undefined;
     fi:
  end proc:

  k := proc(x)
     if is(x, numeric) then

        if x <2 then
            x+1;
        elif x > 2 then
            x+1;
        else
            undefined;
        fi:
     else
        'k(x)';
     end if:
  end proc:

  m := proc(x)
     option Copyright;             # HIDE PROCEDURE CONTENTS!
     if type(x, realcons) then     # SNEAKY TRICK- to prohibit
                                   # symbolic examination of function
                                   # BUT permit plotting!
        evalf(piecewise( x<=1.9995, 2*x+1,
             x>1.9995 and x<=1.99995, 4,
             x>1.99995 and x<=2.00005, 3,
             x>2.00005 and x<=2.0005, 4,
             x>2.0005,2*x+1,undefined));
     else
        'm(x)';
     fi:
  end proc:
end module:
```

Maple Module – CalcToolbox

This file will create a library of useful procedures and functions for use with the calculus labs.

```
> CalcToolbox := module()
   option package;
   export const, dePlot, lim, limitTable, simpleLimitTable, rightLim,
leftLim;

   # CONST() - Define a constant function.  This function is necessary
   # as Maple's limit procedure does not produce correct results when
   # the function is constant.
   # i.e. try
   # evalf(Limit( 1, x=0))
   # doesn't work when limiting point is approached along a constant.

   const := c -> (x-> c + 10.0^(-15) * x);

   dePlot := proc( func1, L, e, a, d)
      # declare any necessary local variables here
      #
      local p1, f1, f2, p2, L1, eRect, dRect, f;

      interface(warnlevel=0):
      if type(func1, procedure) then
         f := func1;
      else
         f := unapply(func1, x);
      end if:

      # include graphics tools (display, and rectangle)
      #
      with( plottools ):  with( plots ):

      # Shade the portion of the graph in the delta band. TRICKY!
      # create a piecewise function the only returns a value when the
      # y-value is within the specified range.

      f1 := y -> piecewise( y<a-d or  y>a+d, f(y), undefined);
      f2 := y -> piecewise( y>=a-d and y<=a+d, f(y), undefined);

      # Note- plot with discont=true so that shading is consistent.
      #
      p1 := plot( f1(x), x=a-2*d..a+2*d, y=L-4*e..L+4*e, discont=true,
thickness=1);
      p2 := plot( f2(x), x=a-2*d..a+2*d, thickness=2,color=blue,
discont=true);

      # add the line y=L
      #
      L1 := plottools[line]( [ a-2*d, L], [ a+2*d, L], color=blue,
linestyle=DASH);
```

```
      # add the epsilon-rectangle
      eRect := plottools[rectangle]( [a-2*d, L+e], [a+2*d, L-
e],color=green);

      # add the delta-rectangle
      #
      dRect := plottools[rectangle]( [a-d,L+4*e],[a+d,L-4*e],
color=yellow);

      # display the graph and the line
      #

      interface(warnlevel=3):
      plots[display]( [ p1, p2, L1, eRect, dRect] );
   end proc:

   # LIMIT - lim( ) procedure
   #  This procedure is to be used by the programming group. This
   # permits the computation of limits of piecewise functions defined
   # as procedures.
   # FOR FINITE LIMITS ON FUNCTIONS DEFINED PROCEDURALLY ONLY!

   lim := proc( )
      option Copyright;          # HIDE PROCEDURE CONTENTS!
      local argList, i, ans, pt, dir, ansLeft,ansRight;

      argList := args[1]+10^(-18)*x;   # Perturb the expression so lim
works for consts!

      dir := 'both';
      for i from 2 to nargs do
          if type( args[i], equation) then  # extract the evaluation
point
              pt := op( 2, args[i] );
          elif args[i] = 'left' then
              dir := 'left';
          elif args[i] = 'right' then
              dir := 'right';
          fi:
          argList := argList,args[i];
       od:

      if dir = 'left' or dir = 'right' then
         # One-sided limit
         #
         ans := round(10^10 * evalf( Limit( argList ),40) ) / 10.^10;
         if not type( ans, realcons) then
            ans := undefined;
         end if:
      else
         # Two-sided limit
         #
         ansLeft  := round( 10^10 * evalf( Limit(argList, left), 40 ) )
/ 10.^10;
         ansRight := round( 10^10 * evalf( Limit(argList, right), 40) )
/ 10.^10;
```

```
         ans := undefined;
         # if either limit is not numeric OR the numeric values differ
         # by more than 10^-6, the limit is declared to not exists.
         #
         if type( ansLeft, realcons) and type( ansRight,realcons) then
            if abs( evalf( ansLeft-ansRight ) ) < 10.0^(-6) then
               ans := ansLeft;
            end if:
         end if:
      end if:

      # if INFINITE, determine sign.
      #
      if ans = Float(infinity) or ans = Float(-infinity) then
         if dir = 'left' then    ans := sign( eval(args[1], x=pt-10^(-
10) ) )*infinity;
         elif dir = 'right' then ans := sign( eval(args[1], x=pt+10^(-
10) ) )*infinity;
         elif ans = Float( infinity) then  ans := infinity;
         elif ans = Float(-infinity) then  ans := -infinity;
         fi:
         ans;
      elif type(ans, realcons) then
         # if finite, then evaluate
         #
         evalf(ans,10);
      else
         ans;
      end if:
   end proc:

   # SimpleLimitTable
   #
   simpleLimitTable := proc(f, x)
      local dMin, d;
      dMin := 0.001:
      d    := 0.1;
      if nargs > 2 then
         dMin := evalf(args[3]);
      end if:

      Digits := 50:
      while d >= dMin do
         printf("%15f %15f\n", x-d, f(x-d));
         d := d / 10;
      end do:

      while d < 0.1 do
         d:= d * 10;
         printf("%15f %15f\n", x+d, f(x+d));
      end do:
   end proc:

   # procedure LimitTable()
   #
   limitTable:= proc(f1, indVar, x, dir)
      option Copyright;          # HIDE PROCEDURE CONTENTS!
```

```
local i, maxI, f;

# Permit the user to specify a function or an expression.
if not type(f1, procedure) then
  f := unapply(f1,indVar);
else
  f := f1;
end if:

maxI := 4;
if nargs > 4 then
   maxI := args[5];
end if:

Digits := 50:
printf( "%15s %15s\n", convert(indVar,string),
convert(f(indVar),string));
printf( "--------------- ---------------\n");

# print sequence of x values approaching from above
#
if dir = right or dir = both then
   for i from 1 by 1 to maxI do
      printf( "%15.6f ", x+10^(-i) );
      try
         if f(x+10^(-i)) <> undefined then
            printf( "%15.6f\n", f(x+10^(-i))  );
         else
            printf( "%15s\n", "UNDEFINED!");
         fi:
      catch:
         printf( "%15s\n", x+10^(-i), "UNDEFINED!"  );
      end try;
   end do:
end if;

printf( "%15.6f ", x );
try
   if f(x) <> undefined then
      printf( "%15.6f\n", f(x) );
   else
      printf( "%15s\n", "UNDEFINED!");
   fi:
catch:
   printf( "%15s\n", "UNDEFINED!" );
end try:

# print sequence of x values approaching from below
#
if dir = left or dir=both then
   for i from maxI by -1 to 1 do
      printf( "%15.6f ", x-10^(-i) );
      try
         if f(x-10^(-i)) <> undefined then
            printf( "%15.6f\n", f(x-10^(-i))  );
         else
            printf( "%15s\n", "UNDEFINED!");
```

```
            fi:
          catch:
            printf( "%15s\n", x-10^(-i), "UNDEFINED!"  );
          end try;
        end do:
      end if:
    end proc:

    rightLim := proc( f1, a, n)
      local i, x1, f;
      if type(f1, procedure) then
        f := f1;
      else
        f := unapply( f1, x);
      end if:

      Digits:=50:
      x1 := 0.1:
      printf("[Right Limit]%2s %15s\n", "x","f(x)");
      printf("--------------- ---------------\n");
      for i from 1 to n do
        printf("%15.8f %15.8f\n", a+x1, f(a+x1));
        x1 := x1/10:
      end do:
      NULL:
    end proc:

    leftLim := proc( f1, a, n)
      local i, x1, f;
      if type(f1, procedure) then
        f := f1;
      else
        f := unapply( f1, x);
      end if:

      Digits := 50:
      x1 := 0.1:
      printf("[Left Limit]%3s %15s\n", "x","f(x)");
      printf("--------------- ---------------\n");
      for i from 1 to n do
        printf("%15.8f %15.8f\n", a-x1, f(a-x1));
        x1 := x1/10:
      end do:
      NULL:
    end proc:

end module:
```