The 30 Year Horizon

Manuel Bronstein  William Burge  Timothy Daly
James Davenport  Michael Dewar  Martin Dunstan
Albrecht Fortenbacher  Patrizia Gianni  Johannes Grabmeier
Jocelyn Guidry  Richard Jenks  Larry Lambe
Michael Monagan  Scott Morrison  William Sit
Jonathan Steinbach  Robert Sutor  Barry Trager
Stephen Watt  Jim Wen  Clifton Williamson

Volume 8.1: Axiom Gallery
Portions Copyright (c) 2005 Timothy Daly

The Blue Bayou image Copyright (c) 2004 Jocelyn Guidry

Portions Copyright (c) 2004 Martin Dunstan
Portions Copyright (c) 2007 Alfredo Portes
Portions Copyright (c) 2007 Arthur Ralfs
Portions Copyright (c) 2005 Timothy Daly

Portions Copyright (c) 1991-2002,
The Numerical ALgorithms Group Ltd.
All rights reserved.

This book and the Axiom software is licensed as follows:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of The Numerical ALgorithms Group Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Inclusion of names in the list of credits is based on historical information and is as accurate as possible. Inclusion of names does not in any way imply an endorsement but represents historical influence on Axiom development.

Michael Albaugh  Cyril Alberga  Roy Adler
Christian Aistleitner  Richard Anderson  George Andrews
S.J. Atkins  Henry Baker  Martin Baker
Stephen Balzac  Yurij Baransky  David R. Barton
Gerald Baumgartner  Gilbert Baumslag  Michael Becker
Nelson H. F. Beebe  Jay Belanger  David Bindel
Fred Blair  Vladimir Bondarenko  Mark Botch
Raoul Bourquin  Alexandre Bouyer  Karen Braman
Peter A. Broadbery  Martin Brock  Manuel Bronstein
Stephen Buchwald  Florian Bundschuh  Luanne Burns
William Burge  Ralph Byers  Quentin Carpent
Robert Caviness  Bruce Char  Ondrej Certik
Tzu-Yi Chen  Cheekai Chin  David V. Chudnovsky
Gregory V. Chudnovsky  Mark Clements  James Cloos
Jia Zhao Cong  Josh Cohen  Christophe Couil
Don Coppersmith  George Corliss  Robert Corless
Gary Cornell  Meino Cramer  Jeremy Du Croz
David Cyganski  Nathaniel Daly  Timothy Daly Sr.
Timothy Daly Jr.  James H. Davenport  David Day
James Demmel  Didier Deshommes  Michael Dewar
Jack Dongarra  Jean Della Dora  Gabriel Dos Reis
Claire DiCrescendo  Sam Dooley  Lionel Ducos
Iain Duff  Lee Duhem  Martin Dunstan
Brian Dupee  Dominique Duval  Robert Edwards
Heow Eide-Goodman  Lars Erickson  Richard Fateman
Bertfried Fauser  Stuart Feldman  John Fletcher
Brian Ford  Albrecht Fortenbacher  George Frances
Constantine Frangos  Timothy Freeman  Korrinn Fu
Marc Gaetano  Rudiger Gebauer  Van de Geijn
Kathy Gerber  Patricia Gianni  Samantha Goldrich
Holger Gollan  Teresa Gomez-Diaz  Laureano Gonzalez-Vega
Stephen Gortler  Johannes Grabmeier  Matt Grayson
Klaus Ebbe Grue  James Griesmer  Vladimir Grinberg
Oswald Gschnitzer  Ming Gu  Jocelyn Guidry
Gaetan Hache  Steve Hague  Satoshi Hamaguchi
Sven Hammarling  Mike Hansen  Richard Hanson
Richard Harke  Bill Hart  Vilya Harvey
Martin Hassner  Arthur S. Hathaway  Dan Hatton
Waldek Hebisch  Karl Hegbloom  Ralf Hemmecke
Henderson Antoine Hersen Roger House
Gernot Hueber
Richard Jenks
Grant Keady
Ted Kosan
Bernhard Kutzler
Kaj Laursen
Frederic Lehobey
Ren-Cang Li
Richard Luczak
Alasdair McAndrew
Edi Meier
Victor S. Miller
H. Michael Moeller
Scott Morrison
William Naylor
John Nelder
Jinzhong Niu
Kostas Oikonomou
Bill Page
Michel Petitot
Frederick H. Pitts
Claude Quitte
Anatoly Rapportirenko
Guilherme Reis
Jean Rivlin
Raymond Rogers
Philip Santas
Gerhard Schneider
Frithjof Schulze
V. Sina
Elena Smirnova
Christine Sundaresan
Eugene Surowitz
James Thatcher
Dylan Thurston
Themos T. Tsikas
Stephen Watt
M. Weller
Thorsten Werther
John M. Wiley
Stephen Wilson
Sandra Wityak
Liu Xiaojun
Vadim Zhytnikov
Bruno Zuercher
Antoine Hersen
Pietro Iglio
William Kahan
Wilfrid Kendall
Paul Kosinski
Tim Lahey
George L. Legendre
Michel Levaud
Rudiger Loos
Cammi Maguire
Bob McElrath
Ian Meikle
Gerard Milmeister
Michael Monagan
Joel Moses
Patrice Naudin
Godfrey Nolan
Michael O’Connor
Humberto Ortiz-Zuazaga
David Parnas
Didier Pinchon
Jose Alfredo Portes
Arthur C. Ralfs
Albert D. Rich
Huan Ren
Nicolas Robidoux
Michael Rothstein
Alfred Scheerhorn
Martin Schoenert
Fritz Schwarz
Nick Simicich
Jonathan Steinbach
Robert Sutor
Max Tegmark
Balbir Thomas
Steve Toleque
Gregory Vanuxem
Jaap Weel
Mark Wegman
Michael Wester
Berhard Will
Shmuel Winograd
Waldemar Wiwianka
Clifford Yapp
Richard Zippel
Dan Zwillinger
Roger House
Alejandro Jakubi
Kai Kaminski
Tony Kennedy
Klaus Kusche
Larry Lambe
Franz Lehner
Howard Levy
Michael Lucks
Francois Maltey
Michael McGetrick
David Mentre
Mohammed Mobarak
Marc Moreno-Maza
Mark Murray
C. Andrew Neff
Arthur Norman
Summat Oemrawsingh
Julian A. Padget
Susan Pelzel
Ayal Pinkus
Gregorio Quintana-Orti
Norman Ramsey
Michael Richardson
Renaud Stumbo
Simon Robinson
Martin Rubey
William Schelter
Marshall Schor
Steven Segletes
William Sit
Fabio Stumbo
Moss E. Sweedler
T. Doug Telford
Mike Thomas
Barry Trager
Bernhard Wall
Juergen Weiss
James Wen
R. Clint Whaley
Clifton J. Williamson
Robert Wisbauer
Knut Wolf
David Yun
Evelyn Zoernack
Contents

1 General examples ............................................ 1
  1.1 Two dimensional functions .............................. 1
    A Simple Sine Function .................................. 2
    A Simple Sine Function, Non-adaptive plot ............. 3
    A Simple Sine Function, Drawn to Scale ................. 4
    A Simple Sine Function, Polar Plot .................... 5
    A Simple Tangent Function, Clipping On ............... 6
    A Simple Tangent Function, Clipping On ............... 7
    Tangent and Sine ....................................... 8
    A 2D Sine Function in BiPolar Coordinates ............ 9
    A 2D Sine Function in Elliptic Coordinates .......... 10
    A 2D Sine Wave in Polar Coordinates .................. 11
  1.2 Two dimensional curves ................................ 11
    A Line in Parabolic Coordinates ....................... 12
    Lissajous Curve ........................................ 13
    A Parametric Curve .................................... 14
    A Parametric Curve in Polar Coordinates .............. 15
  1.3 Three dimensional functions .......................... 15
    A 3D Constant Function in Elliptic Coordinates .... 16
    A 3D Constant Function in Oblate Spheroidal .......... 17
    A 3D Constant in Polar Coordinates ................... 18
    A 3D Constant in Prolate Spheroidal Coordinates ..... 19
    A 3D Constant in Spherical Coordinates ............... 20
    A 2-Equation Space Function ............................ 21
  1.4 Three dimensional curves ................................ 21
    A Parametric Space Curve ................................ 22
    A Tube around a Parametric Space Curve ............... 23
    A 2-Equation Cylindrical Curve ......................... 24
  1.5 Three dimensional surfaces ........................... 24
    A Icosahedron .......................................... 25
    A 3D figure 8 immersion (Klein bagel) ................. 27
    A 2-Equation bipolarCylindrical Surface .............. 28
    A 3-Equation Parametric Space Surface ................ 29
    A 3D Vector of Points in Elliptic Cylindrical ....... 30
CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 3D Constant Function in BiPolar Coordinates</td>
<td>31</td>
</tr>
<tr>
<td>A Swept in Parabolic Coordinates</td>
<td>32</td>
</tr>
<tr>
<td>A Swept Cone in Parabolic Cylindrical Coordinates</td>
<td>33</td>
</tr>
<tr>
<td>A Truncated Cone in Toroidal Coordinates</td>
<td>34</td>
</tr>
<tr>
<td>A Swept Surface in Paraboloidal Coordinates</td>
<td>35</td>
</tr>
<tr>
<td>Jenks Book images</td>
<td>37</td>
</tr>
<tr>
<td>The Complex Gamma Function</td>
<td>38</td>
</tr>
<tr>
<td>The Complex Arctangent Function</td>
<td>39</td>
</tr>
<tr>
<td>Hyperdoc examples</td>
<td>41</td>
</tr>
<tr>
<td>Two dimensional examples</td>
<td>41</td>
</tr>
<tr>
<td>A function of one variable</td>
<td>42</td>
</tr>
<tr>
<td>A Parametric function</td>
<td>43</td>
</tr>
<tr>
<td>A Polynomial in 2 variables</td>
<td>44</td>
</tr>
<tr>
<td>Three dimensional examples</td>
<td>44</td>
</tr>
<tr>
<td>A function of two variables</td>
<td>45</td>
</tr>
<tr>
<td>A parametrically defined curve</td>
<td>46</td>
</tr>
<tr>
<td>A parametrically defined surface</td>
<td>47</td>
</tr>
<tr>
<td>CRC Standard Curves and Surfaces [7]</td>
<td>49</td>
</tr>
<tr>
<td>Standard Curves and Surfaces</td>
<td>49</td>
</tr>
<tr>
<td>CRC graphs</td>
<td>50</td>
</tr>
<tr>
<td>Functions with $x^{n/m}$</td>
<td>50</td>
</tr>
<tr>
<td>Functions with $x^n$ and $(a + bx)^m$</td>
<td>61</td>
</tr>
<tr>
<td>Functions with $a^2 + x^2$ and $x^m$</td>
<td>104</td>
</tr>
<tr>
<td>Functions with $a^2 - x^2$ and $x^m$</td>
<td>114</td>
</tr>
<tr>
<td>Functions with $a^3 + x^3$ and $x^m$</td>
<td>125</td>
</tr>
<tr>
<td>Functions with $a^3 - x^3$ and $x^m$</td>
<td>133</td>
</tr>
<tr>
<td>Functions with $a^4 + x^4$ and $x^m$</td>
<td>140</td>
</tr>
<tr>
<td>Functions with $a^4 - x^4$ and $x^m$</td>
<td>148</td>
</tr>
<tr>
<td>Functions with $(a + bx)^{1/2}$ and $x^m$</td>
<td>156</td>
</tr>
<tr>
<td>Pasta by Design[4]</td>
<td>171</td>
</tr>
<tr>
<td>Acini Di Pepe</td>
<td>172</td>
</tr>
<tr>
<td>Agnolotti</td>
<td>173</td>
</tr>
<tr>
<td>Anellini</td>
<td>174</td>
</tr>
<tr>
<td>Bucatini</td>
<td>175</td>
</tr>
<tr>
<td>Buccoli</td>
<td>176</td>
</tr>
<tr>
<td>Calamaretti</td>
<td>177</td>
</tr>
<tr>
<td>Canneloni</td>
<td>178</td>
</tr>
<tr>
<td>Cannolicchi Rigati</td>
<td>179</td>
</tr>
<tr>
<td>Capellini</td>
<td>180</td>
</tr>
<tr>
<td>Cappelletti</td>
<td>181</td>
</tr>
<tr>
<td>Casarecce</td>
<td>182</td>
</tr>
<tr>
<td>Castellane</td>
<td>183</td>
</tr>
</tbody>
</table>
5.13 Cavatappi ......................................................... 184
5.14 Cavatelli ......................................................... 185
5.15 Chiapperi Rigati .................................................. 186
5.16 Colonie Pompeii .................................................. 187
5.17 Conchiglie Rigate ............................................... 189
5.18 Conchigliette Lisci .............................................. 190
5.19 Conchiglioni Rigate ............................................. 191
5.20 Corallini Lisci .................................................... 192
5.21 Creste Di Galli ..................................................... 193
5.22 Couretti ........................................................... 194
5.23 Ditali Rigati ........................................................ 195
5.24 Fagottini ........................................................... 196
5.25 Farfalle ............................................................ 197
5.26 Farfalline .......................................................... 199
5.27 Farfalloni .......................................................... 200
5.28 Festonati ............................................................ 202
5.29 Fettuccine .......................................................... 203
5.30 Fiocchi Rigati ...................................................... 204
5.31 Fisarmoniche ........................................................ 205
5.32 Funghini ............................................................. 206
5.33 Fusilli ............................................................... 207
5.34 Fusilli al Ferretto .................................................. 208
5.35 Fusilli Capri .......................................................... 209
5.36 Fusilli Lunghi Bucati ............................................. 210
5.37 Galletti .............................................................. 212
5.38 Garganelli ............................................................ 213
5.39 Gemelli .............................................................. 214
5.40 Gigli ................................................................. 215
5.41 Giglio Ondulato ...................................................... 216
5.42 Gnocchetti Sardi ................................................... 217
5.43 Gnocchi ............................................................. 218
5.44 Gramigna ............................................................ 219
5.45 Lancette ............................................................ 220
5.46 Lasagna Larga Doppia Riccia .................................... 221
5.47 Linguine .............................................................. 222
5.48 Lumaconi Rigati ..................................................... 223
5.49 Maccheroni .......................................................... 224
5.50 Maccheroni Alla Chitarra ......................................... 225
5.51 Mafaldine ............................................................ 226
5.52 Manicotti ............................................................. 227
5.53 Orecchiette .......................................................... 229
5.54 Paccheri ............................................................. 230
5.55 Pappardelle .......................................................... 231
5.56 Penne Rigate ........................................................ 232
5.57 Pennoni Lisci ........................................................ 233
5.58 Pennoni Rigati ...................................................... 234
CONTENTS

5.59 Puntalette ................................................................. 235
5.60 Quadrefiore ............................................................... 236
5.61 Quadretti ................................................................. 237
5.62 Racchette ................................................................. 238
5.63 Radiatori ................................................................. 240
5.64 Ravioli Quadrati ......................................................... 241
5.65 Ravioli Tondi .............................................................. 242
5.66 Riccioli ..................................................................... 243
5.67 Riccioli al Cinque Sapori ............................................. 244
5.68 Rigatoni ................................................................. 245
5.69 Rombi ....................................................................... 246
5.70 Rotelle ..................................................................... 247
5.71 Saccottini ................................................................. 248
5.72 Sagnarelli ................................................................. 249
5.73 Sagne Incannulate ...................................................... 250
5.74 Scialatielli ............................................................... 251
5.75 Spaccatelle .............................................................. 252
5.76 Spaghetti ............................................................... 253
5.77 Spiralli ................................................................. 254
5.78 Stellette ................................................................. 255
5.79 Stortini ................................................................. 256
5.80 Strozzapreti ............................................................ 258
5.81 Tagliatelle ............................................................... 259
5.82 Taglierini ............................................................... 260
5.83 Tagliolini ............................................................... 261
5.84 Torchietti .............................................................. 263
5.85 Tortellini ............................................................... 265
5.86 Tortiglioni ............................................................. 266
5.87 Trenne ................................................................. 267
5.88 Tripoline ............................................................... 269
5.89 Trofe ................................................................. 270
5.90 Trottole ............................................................... 271
5.91 Tubetti Rigati .......................................................... 273
5.92 Ziti ................................................................. 274

6 Index ............................................................................. 277
New Foreword

On October 1, 2001 Axiom was withdrawn from the market and ended life as a commercial product. On September 3, 2002 Axiom was released under the Modified BSD license, including this document. On August 27, 2003 Axiom was released as free and open source software available for download from the Free Software Foundation’s website, Savannah.

Work on Axiom has had the generous support of the Center for Algorithms and Interactive Scientific Computation (CAISS) at City College of New York. Special thanks go to Dr. Gilbert Baumslag for his support of the long term goal.

The online version of this documentation is roughly 1000 pages. In order to make printed versions we’ve broken it up into three volumes. The first volume is tutorial in nature. The second volume is for programmers. The third volume is reference material. We’ve also added a fourth volume for developers. All of these changes represent an experiment in print-on-demand delivery of documentation. Time will tell whether the experiment succeeded.

Axiom has been in existence for over thirty years. It is estimated to contain about three hundred man-years of research and has, as of September 3, 2003, 143 people listed in the credits. All of these people have contributed directly or indirectly to making Axiom available. Axiom is being passed to the next generation. I’m looking forward to future milestones.

With that in mind I’ve introduced the theme of the “30 year horizon”. We must invent the tools that support the Computational Mathematician working 30 years from now. How will research be done when every bit of mathematical knowledge is online and instantly available? What happens when we scale Axiom by a factor of 100, giving us 1.1 million domains? How will we integrate theory with code? How will we integrate theorems and proofs of the mathematics with space-time complexity proofs and running code? What visualization tools are needed? How do we support the conceptual structures and semantics of mathematics in effective ways? How do we support results from the sciences? How do we teach the next generation to be effective Computational Mathematicians?

The “30 year horizon” is much nearer than it appears.

Tim Daly
CAISS, City College of New York
November 10, 2003 ((iHy))
Chapter 1

General examples

These examples come from code that ships with Axiom in various input files.

1.1 Two dimensional functions
A Simple Sine Function

draw(sin(11*x), x = 0..2*%pi)
A Simple Sine Function, Non-adaptive plot

draw(sin(11*x), x = 0..2*%pi, adaptive == false, title == "Non-adaptive plot")
A Simple Sine Function, Drawn to Scale

\[
\text{draw}(\sin(11x), x = 0..2\pi, \text{toScale} = \text{true}, \text{title} = \text{"Drawn to scale"})
\]
A Simple Sine Function, Polar Plot

draw(sin(11*x), x = 0..2*%pi, coordinates == polar, title == "Polar plot")
A Simple Tangent Function, Clipping On

draw(tan x, x = -6..6, title == "Clipping on")
A Simple Tangent Function, Clipping On

draw(tan x, x = -6..6, clip == false, title == "Clipping off")
Tangent and Sine

\[ f(x:\text{DFLOAT}) : \text{DFLOAT} = \sin(\tan(x)) - \tan(\sin(x)) \]
\[ \text{draw}(f, 0..6) \]
A 2D Sine Function in BiPolar Coordinates

draw(sin(x),x=0.5..%pi,coordinates == bipolar(1$DFLOAT))
A 2D Sine Function in Elliptic Coordinates

draw(sin(4*t/7), t=0..14*%pi, coordinates == elliptic(1$DFLOAT))
A 2D Sine Wave in Polar Coordinates

\[ \text{draw}(\sin(4t/7), t=0..14\pi, \text{coordinates} = \text{polar}) \]

1.2 Two dimensional curves
A Line in Parabolic Coordinates

\[ h_1(t: \text{DFLOAT}): \text{DFLOAT} == t \]
\[ h_2(t: \text{DFLOAT}): \text{DFLOAT} == 2 \]
\[ \text{draw}(\text{curve}(h_1,h_2),-3..3,\text{coordinates} == \text{parabolic}) \]
Lissajous Curve

\begin{align*}
    i_1(t: FLOAT) & := 9 \sin \left( \frac{3t}{4} \right) \\
    i_2(t: FLOAT) & := 8 \sin (t) \\
    \text{draw(curve(i1, i2), -4*\pi..4*\pi, toScale == true, title == "Lissajous Curve")}
\end{align*}
A Parametric Curve

\[
draw(curve(sin(5*t),t),t = 0..2*\pi,title == "Parametric curve")
\]
A Parametric Curve in Polar Coordinates

\[
\text{draw(curve(sin(5*t),t),t = 0..2*%pi,}
\text{ coordinates == polar,title == "Parametric polar curve")}
\]

1.3 Three dimensional functions
A 3D Constant Function in Elliptic Coordinates

m(u:DFLOAT,v:DFLOAT):DFLOAT == 1
draw(m,0..2*%pi,0..%pi,coordinates == elliptic(1$DFLOAT))
A 3D Constant Function in Oblate Spheroidal

\[ m(u: \text{DFLOAT}, v: \text{DFLOAT}) : \text{DFLOAT} = 1 \]

\[
\text{draw}(m, -\frac{\pi}{2}..\frac{\pi}{2}, 0..2*\pi, \text{coordinates} = \text{oblateSpheroidal}(1 \text{DFLOAT}))
\]
A 3D Constant in Polar Coordinates

\[
m(u: \text{DFLOAT}, v: \text{DFLOAT}): \text{DFLOAT} = 1
\]
\[
draw(m, 0..2*\pi, 0..\pi, \text{coordinates} == \text{polar})
\]
A 3D Constant in Prolate Spheroidal Coordinates

\[ m(u: \text{DFLOAT}, v: \text{DFLOAT}) : \text{DFLOAT} = 1 \]
\[ \text{draw}(m, -\pi/2..\pi/2, 0..2*\pi, \text{coordinates} = \text{prolateSpheroidal}(1\text{DFLOAT})) \]
A 3D Constant in Spherical Coordinates

\[
m(u, v) := 1
\]

\[
draw(m, 0..2*\pi, 0..\pi, \text{coordinates} == \text{spherical})
\]
A 2-Equation Space Function

\[ f(x, y) = \cos(x \cdot y) \]
\[ \text{colorFxn}(x, y) = \frac{1}{x^2 + y^2 + 1} \]
\[ \text{draw}(f, -3..3, -3..3, \text{colorFunction} = \text{colorFxn}, \text{title} = \text{"2-Equation Space Curve"}) \]

1.4 Three dimensional curves
A Parametric Space Curve

\[ \text{draw(curve(sin(t) \cdot \cos(3t/5),\cos(t) \cdot \cos(3t/5),\cos(t) \cdot \sin(3t/5)),}_t = 0..15 \pi, \text{title == "Parametric curve"}) \]
A Tube around a Parametric Space Curve

draw(curve(sin(t)*cos(3*t/5),cos(t)*cos(3*t/5),cos(t)*sin(3*t/5)),
    t = 0..15*%pi,tubeRadius == .15,title == "Tube around curve")

draw(curve(sin(t)*cos(3*t/5),cos(t)*cos(3*t/5),cos(t)*sin(3*t/5)),
    t = 0..15*%pi,tubeRadius == .15,title == "Tube around curve")
A 2-Equation Cylindrical Curve

\begin{equation}
\begin{aligned}
j_1(t:\text{FLOAT}):\text{FLOAT} &= 4 \\
j_2(t:\text{FLOAT}):\text{FLOAT} &= t
\end{aligned}
\end{equation}

draw(curve(j1,j2,j2),-9..9,coordinates == cylindrical)

1.5 Three dimensional surfaces
A Icosahedron

\begin{verbatim}
)se exp add con InnerTrigonometricManipulations
exp(%i*2*%pi/5)
FG2F %
% -1
complexForm %
norm %
simplify %
s:=sqrt %
ph:=exp(%i*2*%pi/5)
A1:=complex(1,0)
A2:=A1*ph
A3:=A2*ph
A4:=A3*ph
A5:=A4*ph
da1:=map(numeric , complexForm FG2F simplify A1)
da2:=map(numeric , complexForm FG2F simplify A2)
da3:=map(numeric , complexForm FG2F simplify A3)
da4:=map(numeric , complexForm FG2F simplify A4)
da5:=map(numeric , complexForm FG2F simplify A5)
B1:=A1*exp(2*%i*%pi/10)
B2:=B1*ph
B3:=B2*ph
B4:=B3*ph
B5:=B4*ph
db1:=map (numeric ,complexForm FG2F simplify B1)
db2:=map (numeric ,complexForm FG2F simplify B2)
db3:=map (numeric ,complexForm FG2F simplify B3)
\end{verbatim}
cb4 := map (numeric , complexForm FG2F simplify B4)
cb5 := map (numeric , complexForm FG2F simplify B5)
u := numeric sqrt(s*s-1)
p0 := point([0,0,u+1/2])@Point(SF)
p1 := point([real ca1,imag ca1,0.5])@Point(SF)
p2 := point([real ca2,imag ca2,0.5])@Point(SF)
p3 := point([real ca3,imag ca3,0.5])@Point(SF)
p4 := point([real ca4,imag ca4,0.5])@Point(SF)
p5 := point([real ca5,imag ca5,0.5])@Point(SF)
p6 := point([real cb1,imag cb1,-0.5])@Point(SF)
p7 := point([real cb2,imag cb2,-0.5])@Point(SF)
p8 := point([real cb3,imag cb3,-0.5])@Point(SF)
p9 := point([real cb4,imag cb4,-0.5])@Point(SF)
p10 := point([real cb5,imag cb5,-0.5])@Point(SF)
p11 := point([0,0,-u-1/2])@Point(SF)
space := create3Space()$ThreeSpace DFloat
polygon(space, [p0,p1,p2])
polygon(space, [p0,p2,p3])
polygon(space, [p0,p3,p4])
polygon(space, [p0,p4,p5])
polygon(space, [p0,p5,p6])
polygon(space, [p1,p6,p2])
polygon(space, [p1,p7,p3])
polygon(space, [p2,p7,p4])
polygon(space, [p3,p8,p4])
polygon(space, [p4,p9,p5])
polygon(space, [p5,p10,p1])
polygon(space, [p5,p11,p2])
polygon(space, [p6,p11,p7])
polygon(space, [p3,p7,p8])
polygon(space, [p4,p8,p9])
polygon(space, [p5,p9,p10])
polygon(space, [p1,p10,p6])
polygon(space, [p6,p11,p7])
polygon(space, [p7,p11,p8])
polygon(space, [p8,p11,p9])
polygon(space, [p9,p11,p10])
polygon(space, [p10,p11,p6])
makeViewport3D(space, title="Icosahedron", style="smooth")
1.5. THREE DIMENSIONAL SURFACES

A 3D figure 8 immersion (Klein bagel)

From en.wikipedia.org/wiki/Klein_bottle. The “figure 8” immersion (Klein bagel) of the Klein bottle has a particularly simple parameterization. It is that of a “figure 8” torus with a 180 degree “Möbius” twist inserted. In this immersion, the self-intersection circle is a geometric circle in the x-y plane. The positive constant \( r \) is the radius of this circle. The parameter \( u \) gives the angle in the x-y plane, and \( v \) specifies the position around the 8-shaped cross section. With the above parameterization the cross section is a 2:1 Lissajous curve.
A 2-Equation bipolarCylindrical Surface

\[ u \times \cos(v) \]

\[
\text{draw(surface}(u \times \cos(v), u \times \sin(v), u), u=1..4, v=1..2\pi, _
\text{coordinates == bipolarCylindrical(1\$DFLOAT)))}
\]
A 3-Equation Parametric Space Surface

\[
\begin{align*}
n_1(u, v) & = u \cos(v) \\
n_2(u, v) & = u \sin(v) \\
n_3(u, v) & = v \cos(u) \\
colorFxn(x, y) & = \frac{1}{x^2 + y^2 + 1}
\end{align*}
\]

draw(surface(n1, n2, n3), -4..4, 0..2*%pi, colorFunction == colorFxn)
A 3D Vector of Points in Elliptic Cylindrical

\[
\begin{align*}
\text{vector} & \quad \text{[0,0,1]} \\
\beta(u) & \quad \text{vector} \quad [\cos u, \sin u, 0] \\
\delta(u) & \quad (\cos(u/2)) \times \beta(u) + \sin(u/2) \times \text{U2} \\
\text{vec} & \quad x(u,v) = \beta(u) + v \times \delta(u) \\
\end{align*}
\]

\[\text{draw(surface(vec.1,vec.2,vec.3),v=-0.5..0.5,u=0..2*%pi,coordinates == ellipticCylindrical(1$DFLOAT),var1Steps == 50,var2Steps == 50)}\]
1.5. THREE DIMENSIONAL SURFACES

A 3D Constant Function in BiPolar Coordinates

\[ m(u: \text{FLOAT}, v: \text{FLOAT}): \text{FLOAT} \equiv 1 \]
\[ \text{draw}(m, 0..2\pi, 0..\pi, \text{coordinates} == \text{bipolar}(1\text{FLOAT})) \]
A Swept in Parabolic Coordinates

\[ u \times \cos(v) \]

\[
\text{draw(surface}(u \times \cos(v), u \times \sin(v), 2u), u=0..4, v=0..2\pi, \text{coordinates==parabolic})
\]

— equation115 —
A Swept Cone in Parabolic Cylindrical Coordinates

\[ u \cos(v) \]

\[
draw(surface(u*\cos(v),u*\sin(v),v*\cos(u)),u=0..4,v=0..2*%pi,\
coordinates == parabolicCylindrical)
\]
A Truncated Cone in Toroidal Coordinates

\[ u \cdot \cos(v) \]

```latex
\begin{verbatim}
  draw(surface(u*cos(v),u*sin(v),u),u=1..4,v=1..4*%pi,
       coordinates == toroidal(1$DFLOAT))
\end{verbatim}
```
A Swept Surface in Paraboloidal Coordinates

\[ u \cdot \cos(v) \]

--- equation117 ---

\[
\text{draw(surface}(u\cos(v),u\sin(v),u*v),u=0..4,v=0..2*\pi,\quad
\text{coordinates==paraboloidal,steps u== 50, steps v== 50})
\]
Chapter 2

Jenks Book images
The Complex Gamma Function

\[ \Gamma(x + iy) \]

A 3-d surface whose height is the real part of the Gamma function, and whose color is the argument of the Gamma function.
The Complex Arctangent Function

```
| complexarctangent |

---

```

The complex arctangent function. The height is the real part and the color is the argument.
Chapter 3

Hyperdoc examples

Examples in this section come from the Hyperdoc documentation tool. These examples are accessed from the Basic Examples Draw section.

3.1 Two dimensional examples
A function of one variable

\[ y = f(x) \]

where \( y \) is the dependent variable and \( x \) is the independent variable.
A Parametric function

draw(curve(-9*sin(4*t/5),8*sin(t)), t=-5*%pi..5*%pi, title="Lissajous")

This is one of the demonstration equations used in hypertex. It draw a parametrically defined curve

\[ f_1(t), f_2(t) \]

in terms of two functions \( f_1 \) and \( f_2 \) and an independent variable \( t \).
A Polynomial in 2 variables

\[ y^2 + 7xy - (x^3 + 16x) = 0, \]

This is one of the demonstration equations used in hypertex. Plotting the solution to

\[ p(x, y) = 0 \]

where \( p \) is a polynomial in two variables \( x \) and \( y \).

### 3.2 Three dimensional examples
A function of two variables

\[ D \exp(\log(x+y) + \cos(y-x)) = 2 \]

This is one of the demonstration equations used in hypertex. A function of two variables

\[ z = f(x, y) \]

where \( z \) is the dependent variable and where \( x \) and \( y \) are the dependent variables.
A parametrically defined curve

\( f_1(t) = 1.3 \cos(2t) \cos(4t) + \sin(4t) \cos(t) \)
\( f_2(t) = 1.3 \sin(2t) \cos(4t) - \sin(4t) \sin(t) \)
\( f_3(t) = 2.5 \cos(4t) \)

\( cf(x,y) = 0.5 \)

\[
\text{draw(curve}(f_1(t),f_2(t),f_3(t)),t=0..4\%\pi,\text{tubeRadius}=.25,\text{tubePoints}=16,\_ \\
\hspace{1cm}\text{title}=="knot","\text{colorFunction}==cf,\text{style}=="\text{smooth}"
\]

This is one of the demonstration equations used in hypertex. This ia parametrically defined curve

\( f_1(t), f_2(t), f_3(t) \)

in terms of three functions \( f_1, f_2, \) and \( f_3 \) and an independent variable \( t \).
A parametrically defined surface

\begin{equation}
\begin{align*}
f_1(u,v) &= u \sin(v) \\
f_2(u,v) &= v \cos(u) \\
f_3(u,v) &= u \cos(v)
\end{align*}
\end{equation}

draw(surface(f_1(u,v), f_2(u,v), f_3(u,v)), u=-\pi..\pi, v=-\pi/2..\pi/2, \\
\quad title=="surface", colorFunction==cf, style=="smooth")

This is one of the demonstration equations used in hypertex. This is a parametrically defined curve

\begin{equation}
f_1(t), f_2(t), f_3(t)
\end{equation}

in terms of three functions $f_1$, $f_2$, and $f_3$ and an independent variable $t$. 
Chapter 4

CRC Standard Curves and Surfaces [7]

4.1 Standard Curves and Surfaces

In order to have an organized and thorough evaluation of the Axiom graphics code we turn to the CRC Standard Curves and Surfaces (SCC). This volume was written years after the Axiom graphics code was written so there was no attempt to match the two until now. However, the SCC volume will give us a solid foundation to both evaluate the features of the current code and suggest future directions.

According to the SCC we can organize the various curves by the taxonomy:

1 random
   1.1 fractal
   1.2 gaussian
   1.3 non-gaussian

2 determinate
   2.1 algebraic – A polynomial is defined as a summation of terms composed of integral powers of $x$ and $y$. An algebraic curve is one whose implicit function

   \[ f(x, y) = 0 \]

   is a polynomial in $x$ and $y$ (after rationalization, if necessary). Because a curve is often defined in the explicit form

   \[ y = f(x) \]

   there is a need to distinguish rational and irrational functions of $x$. 

49
2.1.1 **irrational** – An irrational function of \( x \) is a quotient of two polynomials, one or both of which has a term (or terms) with power \( p/q \), where \( p \) and \( q \) are integers.

2.1.2 **rational** – A rational function of \( x \) is a quotient of two polynomials in \( x \), both having only integer powers.

2.1.2.1 **polynomial**

2.1.2.2 **non-polynomial**

2.2 **integral** – Certain continuous functions are not expressible in algebraic or transcendental forms but are familiar mathematical tools. These curves are equal to the integrals of algebraic or transcendental curves by definition; examples include Bessel functions, Airy integrals, Fresnel integrals, and the error function.

2.3 **transcendental** – The transcendental curves cannot be expressed as polynomials in \( x \) and \( y \). These are curves containing one or more of the following forms: exponential (\( e^x \)), logarithmic (\( \log(x) \)), or trigonometric (\( \sin(x) \), \( \cos(x) \)).

2.3.1 **exponential**

2.3.2 **logarithmic**

2.3.3 **trigonometric**

2.4 **piecewise continuous** – Other curves, except at a few singular points, are smooth and differentiable. The class of nondifferentiable curves have discontinuity of the first derivative as a basic attribute. They are often composed of straight-line segments. Simple polygonal forms, regular fractal curves, and turtle tracks are examples.

2.4.1 **periodic**

2.4.2 **non-periodic**

2.4.3 **polygonal**

2.4.3.1 **regular**

2.4.3.2 **irregular**

2.4.3.3 **fractal**

4.2 **CRC graphs**

**Functions with** \( x^{n/m} \)

Page 26 2.1.1

\[
y = cx^n \\
y - cx^n = 0
\]
4.2. CRC GRAPHS

)clear all
)set mes auto off
f(c,x,n) == c*x^n
lineColorDefault(green())
viewport1:=draw(f(1,x,1), x=-2..2, adaptive==true, unit==[1.0,1.0],
   title=="p26-2.1.1.1-3")
graph2111:=getGraph(viewport1,1)

lineColorDefault(blue())
viewport2:=draw(f(1,x,3), x=-2..2, adaptive==true, unit==[1.0,1.0])
graph2112:=getGraph(viewport2,1)

lineColorDefault(red())
viewport3:=draw(f(1,x,5), x=-2..2, adaptive==true, unit==[1.0,1.0])
graph2113:=getGraph(viewport3,1)

putGraph(viewport1,graph2112,2)
putGraph(viewport1,graph2113,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\begin{verbatim}
)clear all
)set mes auto off
f(c,x,n) == c*x^n
lineColorDefault(green())
viewport1:=draw(f(1,x,2), x=-2..2, adaptive==true, unit==[1.0,1.0],
   title=="p26-2.1.1.4-6")
graph2111:=getGraph(viewport1,1)

lineColorDefault(blue())
viewport2:=draw(f(1,x,4), x=-2..2, adaptive==true, unit==[1.0,1.0])
graph2112:=getGraph(viewport2,1)

lineColorDefault(red())
viewport3:=draw(f(1,x,6), x=-2..2, adaptive==true, unit==[1.0,1.0])
graph2113:=getGraph(viewport3,1)

putGraph(viewport1,graph2112,2)
putGraph(viewport1,graph2113,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\end{verbatim}
4.2. CRC GRAPHS

Page 26 2.1.2

\[ y = \frac{c}{x^n} \]

\[ yx^n - c = 0 \]

--- p26-2.1.2.1-3 ---

)clear all
f(c,x,n) := c/x^n
lineColorDefault(green())
viewport1:=draw(f(0.01,x,1),x=-2..2,adaptive==true,unit=[[1.0,1.0],
        title=="p26-2.1.2.1-3")
graph2111:=getGraph(viewport1,1)
lineColorDefault(blue())
viewport2:=draw(f(0.01,x,3),x=-4..4,adaptive==true,unit=[[1.0,1.0])
graph2122:=getGraph(viewport2,1)
lineColorDefault(red())
viewport3:=draw(f(0.01,x,5),x=-2..2,adaptive==true,unit=[[1.0,1.0])
graph2123:=getGraph(viewport3,1)
putGraph(viewport1,graph2122,2)
putGraph(viewport1,graph2123,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

---
Note that Axiom’s plot of viewport2 disagrees with CRC

---

```plaintext

```
4.2. CRC GRAPHS

\[ y = cx^{n/m} \]
\[ y - cx^{n/m} = 0 \]

-- p28-2.1.3.1-6 --

)clear all  
f(c,x,n,m) == c*x^(n/m)

lineColorDefault(color(1))  
viewport1:=draw(f(1,x,1,4),x=0..1,adaptive=true,unit=[[1.0,1.0]],
    title="p28-2.1.3.1-6")
graph2131:=getGraph(viewport1,1)

lineColorDefault(color(2))  
viewport2:=draw(f(1,x,1,2),x=0..1,adaptive=true,unit=[[1.0,1.0]])
graph2132:=getGraph(viewport2,1)

lineColorDefault(color(3))  
viewport3:=draw(f(1,x,3,4),x=0..1,adaptive=true,unit=[[1.0,1.0]])
graph2133:=getGraph(viewport3,1)

lineColorDefault(color(4))  
viewport4:=draw(f(1,x,5,4),x=0..1,adaptive=true,unit=[[1.0,1.0]])
CHAPTER 4. CRC STANDARD CURVES AND SURFACES

```
graph2134 := getGraph(viewport4, 1)
lineColorDefault(color(5))
viewport5 := draw(f(1, x, 3, 2), x=0..1, adaptive=true, unit=[1.0, 1.0])
graph2135 := getGraph(viewport5, 1)
lineColorDefault(color(6))
viewport6 := draw(f(1, x, 7, 4), x=0..1, adaptive=true, unit=[1.0, 1.0])
graph2136 := getGraph(viewport6, 1)
putGraph(viewport1, graph2132, 2)
putGraph(viewport1, graph2133, 3)
putGraph(viewport1, graph2134, 4)
putGraph(viewport1, graph2135, 5)
putGraph(viewport1, graph2136, 6)
units(viewport1, 1, "on")
points(viewport1, 1, "off")
points(viewport1, 2, "off")
points(viewport1, 3, "off")
makeViewport2D(viewport1)
```
4.2. CRC GRAPHS

\[ f(c,x,n,m) = c \times x^{\frac{n}{m}} \]

`lineColorDefault(color(1))`  
`viewport1:=draw(f(1,x,1,3),x=0..1,adaptive=true,unit=[1.0,1.0], _  
  title="p28-2.1.3.7-10")`  
`graph2137:=getGraph(viewport1,1)`

`lineColorDefault(color(2))`  
`viewport2:=draw(f(1,x,2,3),x=0..1,adaptive=true,unit=[1.0,1.0])`  
`graph2138:=getGraph(viewport2,1)`

`lineColorDefault(color(3))`  
`viewport3:=draw(f(1,x,4,3),x=0..1,adaptive=true,unit=[1.0,1.0])`  
`graph2139:=getGraph(viewport3,1)`

`lineColorDefault(color(4))`  
`viewport4:=draw(f(1,x,5,3),x=0..1,adaptive=true,unit=[1.0,1.0])`  
`graph21310:=getGraph(viewport4,1)`

`putGraph(viewport1,graph2138,2)`  
`putGraph(viewport1,graph2139,3)`  
`putGraph(viewport1,graph21310,4)`  
`units(viewport1,"on")`  
`points(viewport1,1,"off")`  
`points(viewport1,2,"off")`  
`points(viewport1,3,"off")`  
`makeViewport2D(viewport1)`
\[ y = \frac{c}{x^{n/m}} \]

\[ yx^{n/m} - c = 0 \]

__)clear all
f(c,x,n,m) == c/x^(n/m)

lineColorDefault(color(1))
viewport1:=draw(f(0.01,x,1,4),x=0..1,adaptive=true,unit=[1.0,1.0],
   title="p28-2.1.4.1-6")
graph2141:=getGraph(viewport1,1)

lineColorDefault(color(2))
viewport2:=draw(f(0.01,x,1,2),x=0..1,adaptive=true,unit=[1.0,1.0])
graph2142:=getGraph(viewport2,1)

lineColorDefault(color(3))
viewport3:=draw(f(0.01,x,3,4),x=0..1,adaptive=true,unit=[1.0,1.0])
graph2143:=getGraph(viewport3,1)

lineColorDefault(color(4))
4.2. CRC GRAPHS

viewport4:=draw(f(0.01,x,5,4),x=0..1,adaptive==true,unit==[1.0,1.0])
graph2144:=getGraph(viewport4,1)

lineColorDefault(color(5))
viewport5:=draw(f(0.01,x,3,2),x=0..1,adaptive==true,unit==[1.0,1.0])
graph2145:=getGraph(viewport5,1)

lineColorDefault(color(6))
viewport6:=draw(f(0.01,x,7,4),x=0..1,adaptive==true,unit==[1.0,1.0])
graph2146:=getGraph(viewport6,1)

putGraph(viewport1,graph2142,2)
putGraph(viewport1,graph2143,3)
putGraph(viewport1,graph2144,4)
putGraph(viewport1,graph2145,5)
putGraph(viewport1,graph2146,6)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

---

p28-2.1.4.1-6

--- p28-2.1.4.7-10 ---

)clear all
\[ f(c,x,n,m) = \frac{c}{x^{(n/m)}} \]

```
lineColorDefault(color(1))
viewport1:=draw(f(1,x,1,3),x=0..1,adaptive==true,unit==[1.0,1.0],
               title=="p28-2.1.4.7-10")
graph2147:=getGraph(viewport1,1)

lineColorDefault(color(2))
viewport2:=draw(f(1,x,2,3),x=0..1,adaptive==true,unit==[1.0,1.0])
graph2148:=getGraph(viewport2,1)

lineColorDefault(color(3))
viewport3:=draw(f(1,x,4,3),x=0..1,adaptive==true,unit==[1.0,1.0])
graph2149:=getGraph(viewport3,1)

lineColorDefault(color(4))
viewport4:=draw(f(1,x,5,3),x=0..1,adaptive==true,unit==[1.0,1.0])
graph21410:=getGraph(viewport4,1)

putGraph(viewport1,graph2148,2)
putGraph(viewport1,graph2149,3)
putGraph(viewport1,graph21410,4)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```
Functions with $x^n$ and $(a + bx)^m$

Page 30 2.2.1

\[ y = c(a + bx) \]

\[ y - bcx - ac = 0 \]
$y = c(a + bx)^2$

$y - cb^2x^2 - 2abcx - a^2c = 0$

— p30-2.2.1-3 —
4.2. CRC GRAPHS

\[
y = c(a + bx)^3
\]

\[
y - b^3cx^3 - 3ab^2cx^2 - 3a^2bcx - a^3c = 0
\]
clear all

\( f(x,a,b,c) = c \cdot (a+b \cdot x)^3 \)

lineColorDefault(red())
viewport1 := draw(f(x,0.5,0.5,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0],
  title="p30-2.2.3.1-3")
graph2231 := getGraph(viewport1,1)

lineColorDefault(green())
viewport2 := draw(f(x,0.5,1.0,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0])
graph2232 := getGraph(viewport2,1)

lineColorDefault(blue())
viewport3 := draw(f(x,0.5,2.0,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0])
graph2233 := getGraph(viewport3,1)

putGraph(viewport1,graph2232,2)
putGraph(viewport1,graph2233,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\[ y = cx(a + bx) \]
\[ y - bcx^2 - acx = 0 \]
\[ y = cx(a + bx)^2 \]
\[ y - b^2cx^3 - 2abcx^2 - a^2cx = 0 \]
4.2. CRC GRAPHS

\[ y = cx(a + bx)^3 \]
\[ y - b^3cx^3 - 3ab^2cx - 3a^2bx^2 - a^3cx = 0 \]
\texttt{clear all}
\texttt{f(x,a,b,c) == c*x*(a+b*x)^3}

\texttt{lineColorDefault(red())}
\texttt{viewport1:=draw(f(x,0.5,0.5,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0],-_title="p30-2.2.6.1-3")}
\texttt{graph2261:=getGraph(viewport1,1)}

\texttt{lineColorDefault(green())}
\texttt{viewport2:=draw(f(x,0.5,1.0,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0])}
\texttt{graph2262:=getGraph(viewport2,1)}

\texttt{lineColorDefault(blue())}
\texttt{viewport3:=draw(f(x,0.5,2.0,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0])}
\texttt{graph2263:=getGraph(viewport3,1)}

\texttt{putGraph(viewport1,graph2262,2)}
\texttt{putGraph(viewport1,graph2263,3)}
\texttt{units(viewport1,1,"on")}
\texttt{points(viewport1,1,"off")}
\texttt{points(viewport1,2,"off")}
\texttt{points(viewport1,3,"off")}
\texttt{makeViewport2D(viewport1)}
\[ y = cx^2(a + bx) \]
\[ y - bcx^3 - acx^2 = 0 \]

--- p30-2.2.7.1-3 ---

```plaintext
)clear all
f(x,a,b,c) == c*x^2*(a+b*x)

lineColorDefault(red())
viewport1:=draw(f(x,0.5,0.5,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0],
title="p30-2.2.7.1-3")
graph2271:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,1.0,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0])
graph2272:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.5,2.0,1.0),x=-2..1,adaptive=true,unit=[1.0,1.0])
graph2273:=getGraph(viewport3,1)

putGraph(viewport1,graph2272,2)
putGraph(viewport1,graph2273,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```

---
\[ y = cx^2(a + bx)^2 \]

\[ y - b^2cx^4 - 2abcx^3 - a^2cx^2 = 0 \]
4.2. CRC GRAPHS

units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

$y = cx^2(a + bx)^3$

$y - b^3cx^5 - 3ab^2cx^4 - 3a^2bcx^3 - a^3cx^2 = 0$

)clear all
f(x,a,b,c) == c*x^2*(a+b*x)^3

lineColorDefault(red())
viewport1:=draw(f(x,0.5,0.5,1.0),x=-2..1,adaptive==true,unit==[1.0,1.0],_
title="p32-2.2.9.1-3")
graph2291:=getGraph(viewport1,1)
y = cx^3(a + bx)

y - bcx^4 - acx^3 = 0
4.2. CRC GRAPHS

)clear all
f(x,a,b,c) == c*x^3*(a+b*x)

lineColorDefault(red())
viewport1:=draw(f(x,0.5,0.5,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0],_
   title=="p32-2.2.10.1-3")
graph22101:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,1.0,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22102:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.5,2.0,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22103:=getGraph(viewport3,1)

putGraph(viewport1,graph22102,2)
putGraph(viewport1,graph22103,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

---
\[ y = cx^3(a + bx)^2 \]
\[ y - b^2cx^5 - 2abcx^4 - a^2cx^3 = 0 \]

--- p32-2.2.11.1-3 ---

```plaintext
)clear all
f(x,a,b,c) == c*x^3*(a+b*x)^2

lineColorDefault(red())
viewport1:=draw(f(x,0.5,0.5,1.0),x=-2..2,adaptive=true,unit=[1.0,1.0],
    title="p32-2.2.11.1-3")
graph22111:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,1.0,1.0),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph22112:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.5,2.0,1.0),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph22113:=getGraph(viewport3,1)

putGraph(viewport1,graph22112,2)
putGraph(viewport1,graph22113,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```
4.2. CRC GRAPHS

\[ y = cx^3(a + bx)^3 \]
\[ y - b^3cx^6 - 3ab^2cx^5 - 3a^2bcx^4 - a^3cx^3 = 0 \]

\[ \text{clear all} \]
\[ f(x,a,b,c) == c*x^3*(a+b*x)^3 \]

\[ \text{lineColorDefault(red())} \]
view1:=draw(f(x,0.5,0.5,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0],_title=="p32-2.2.12.1-3")
graph22121:=getGraph(view1,1)

\[ \text{lineColorDefault(green())} \]
view2:=draw(f(x,0.5,1.0,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22122:=getGraph(view2,1)

\[ \text{lineColorDefault(blue())} \]
view3:=draw(f(x,0.5,2.0,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22123:=getGraph(view3,1)

putGraph(view1,graph22122,2)
putGraph(view1,graph22123,3)
\[ y = \frac{c}{a + bx} \]

\[ ay + bxy - c = 0 \]
\[
y = \frac{c}{(a + bx)^2}
\]
\[
a^2y + 2abxy + b^2x^2y - c = 0
\]
clear all
f(x,a,b,c) == c/(a+b*x)^2

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0],_
    title=="p34-2.2.14.1-3")
graph22141:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,3.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22142:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,4.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22143:=getGraph(viewport3,1)

putGraph(viewport1,graph22142,2)
putGraph(viewport1,graph22143,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
4.2. CRC GRAPHS

Page 34 2.2.15

\[ y = \frac{c}{(a + bx)^3} \]

\[ a^3 y + 2a^2 bxy + 2ab^2 x^2 y + b^3 x^3 y - c = 0 \]

--- p34-2.2.15.1-3 ---

)clear all
f(x,a,b,c) == c/(a+b*x)^3

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0],
   title=="p34-2.2.15.1-3")
graph22151:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,3.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22152:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,4.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22153:=getGraph(viewport3,1)

putGraph(viewport1,graph22152,2)
putGraph(viewport1,graph22153,3)
units(viewport1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

---
\[ y = \frac{cx}{a + bx} \]

\[ ay + bxy - cx = 0 \]

--- p34-2.2.16.1-3 ---

```plaintext
)clear all
f(x,a,b,c) == c*x/(a+b*x)

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.1),x=-2..2,adaptive==true,unit==[1.0,1.0],
    title=="p34-2.2.16.1-3")
graph22161:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,3.0,0.1),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22162:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,4.0,0.1),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22163:=getGraph(viewport3,1)

putGraph(viewport1,graph22162,2)
```
4.2. CRC GRAPHS

\[ y = \frac{cx}{(a + bx)^2} \]

\[ a^2y + 2abx + b^2x^2y - cx = 0 \]
$y = \frac{cx}{(a + bx)^3}$

\[ a^3y + 3a^2bxy + 3ab^2x^2y + b^3x^3y - cx = 0 \]
4.2. CRC GRAPHS

--- p34-2.2.18.1-3 ---

)clear all
f(x,a,b,c) == c*x/(a+b*x)^3

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.01),x=-2..2,adaptive=true,unit=[[1.0,1.0],
title="p34-2.2.18.1-3")
graph22181:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,3.0,0.01),x=-2..2,adaptive=true,unit=[[1.0,1.0])
graph22182:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,4.0,0.01),x=-2..2,adaptive=true,unit=[[1.0,1.0])
graph22183:=getGraph(viewport3,1)

putGraph(viewport1,graph22182,2)
putGraph(viewport1,graph22183,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\[ y = \frac{c x^2}{(a + b x)} \]

\[ a y + b x y - c x^2 = 0 \]
4.2. CRC GRAPHS

\[ y = \frac{cx^2}{(a + bx)^2} \]

\[ a^2y + 2abxy + b^2x^2y - cx^2 = 0 \]
\[ y = \frac{cx^2}{(a + bx)^3} \]

\[ a^3y + 3a^2bxy + 3ab^2x^2y + b^3x^3y - cx^2 = 0 \]
4.2. CRC GRAPHS

— p36-2.2.21.1-3 —

)clear all
f(x,a,b,c) == c*x^2/(a+b*x)^3
lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0],
    title=="p36-2.2.21.1-3")
graph22211:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,3.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22212:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,4.0,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22213:=getGraph(viewport3,1)

putGraph(viewport1,graph22212,2)
putGraph(viewport1,graph22213,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\[ y = \frac{cx^3}{a + bx} \]

\[ ay + bxy - cx^3 = 0 \]

--- p36-2.2.22.1-3 ---

\[ y = \frac{cx^3}{a + bx} \]

\[ ay + bxy - cx^3 = 0 \]
4.2. CRC GRAPHS

\[
\begin{align*}
\text{putGraph(viewport1,graph22223,3)} \\
\text{units(viewport1,1,"on")} \\
\text{points(viewport1,1,"off")} \\
\text{points(viewport1,2,"off")} \\
\text{points(viewport1,3,"off")} \\
\text{makeViewport2D(viewport1)} \\
\end{align*}
\]

\[
\begin{align*}
y &= \frac{cx^3}{(a + bx)^2} \\
a^2y + 2abxy + b^2x^2y - cx^3 &= 0
\end{align*}
\]

-- p36-2.2.23.1-3 --

\`
\text{clear all} \\
f(x,a,b,c) == c*x^3/(a+b*x)^2 \\
\text{lineColorDefault(red())} \\
\text{viewport1:=draw(f(x,1.0,2.0,0.2),x=-2..2,adaptive=true,unit=[1.0,1.0],_} \\
\text{title="p36-2.2.23.1-3")}
\`
$$y = \frac{cx^3}{(a + bx)^3}$$

$$a^3y + 3a^2bxy + 3ab^2x^2y + b^3x^3y - cx^3 = 0$$
4.2. CRC GRAPHS

— p36-2.2.24.1-3 —

\[ f(x, a, b, c) = \frac{c x^3}{(a+b x)^3} \]

\texttt{lineColorDefault(red())}
\texttt{viewport1:=draw(f(x,1.0,2.0,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0],_}
\texttt{title="p36-2.2.24.1-3")}
\texttt{graph22241:=getGraph(viewport1,1)}

\texttt{lineColorDefault(green())}
\texttt{viewport2:=draw(f(x,1.0,3.0,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\texttt{graph22242:=getGraph(viewport2,1)}

\texttt{lineColorDefault(blue())}
\texttt{viewport3:=draw(f(x,1.0,4.0,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\texttt{graph22243:=getGraph(viewport3,1)}

\texttt{putGraph(viewport1,graph22242,2)}
\texttt{putGraph(viewport1,graph22243,3)}
\texttt{units(viewport1,"on")}
\texttt{points(viewport1,"off");}
\texttt{points(viewport1,2,"off")}
\texttt{points(viewport1,3,"off")}
\texttt{makeViewport2D(viewport1)
\[ y = \frac{c(a + bx)}{x} \]

\[ xy - bcx - ca = 0 \]

--- p38-2.2.25.1-3 ---

```plaintext
)clear all
f(x,a,b,c) == c*(a+b*x)/x

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.04),x=-0.5..0.5,adaptive==true,unit==[1.0,1.0],_
   title=="p38-2.2.25.1-3")
graph22251:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,4.0,0.04),x=-0.5..0.5,adaptive==true,unit==[1.0,1.0])
graph22252:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,6.0,0.04),x=-0.5..0.5,adaptive==true,unit==[1.0,1.0])
graph22253:=getGraph(viewport3,1)

putGraph(viewport1,graph22252,2)
```
4.2. CRC GRAPHS

\[ y = \frac{c(a + bx)^2}{x} \]

\[ xy - b^2cx^2 - 2abcx - a^2c = 0 \]

--- p38-2.2.26.1-3 ---

)clear all
f(x,a,b,c) == c*(a+b*x)^2/x

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.04),x=-2..2,adaptive=true,unit=[[1.0,1.0],
    title="p38-2.2.26.1-3")
graph22261:=getGraph(viewport1,1)
\[ y = \frac{c(a + bx)^3}{x} \]

\[ xy - b^3 cx^3 - 3ab^2 cx^2 - 3a^2 b cx - a^3 c = 0 \]
4.2. CRC GRAPHS

— p38-2.2.27.1-3 —

\[
\text{clear all}
\]

\[
f(x, a, b, c) = c \frac{(a + b \times x)^3}{x}
\]

\[
\text{lineColorDefault(red())}
\]

\[
\text{viewport1:=draw(f(x,1.0,2.0,0.02),x=-2..2,adaptive=true,unit=[1.0,1.0],}
\]

\[
\text{title="p38-2.2.27.1-3")}
\]

\[
\text{graph22271:=getGraph(viewport1,1)}
\]

\[
\text{lineColorDefault(green())}
\]

\[
\text{viewport2:=draw(f(x,1.0,4.0,0.02),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\]

\[
\text{graph22272:=getGraph(viewport2,1)}
\]

\[
\text{lineColorDefault(blue())}
\]

\[
\text{viewport3:=draw(f(x,1.0,6.0,0.02),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\]

\[
\text{graph22273:=getGraph(viewport3,1)}
\]

\[
\text{putGraph(viewport1,graph22272,2)}
\]

\[
\text{putGraph(viewport1,graph22273,3)}
\]

\[
\text{units(viewport1,1,"on")}
\]

\[
\text{points(viewport1,1,"off")}
\]

\[
\text{points(viewport1,2,"off")}
\]

\[
\text{points(viewport1,3,"off")}
\]

\[
\text{makeViewport2D(viewport1)}
\]
\[ \begin{align*}
    y &= \frac{c(a + bx)}{x^2} \\
    x^2y - bex - ca &= 0
\end{align*} \]

— p38-2.2.28.1-3 —
4.2. CRC GRAPHS

\[ y = \frac{c(a + bx)^2}{x^2} \]

\[ x^2y - b^2cx^2 - 2abcx - a^2c = 0 \]
\[ y = \frac{c(a + bx)^3}{x^2} \]

\[ x^2 y - b^3 c x^3 - 3ab^2 c x^2 - 3a^2 b c x - a^3 c = 0 \]
4.2. CRC GRAPHS

— p38-2.2.30.1-3 —

\begin{verbatim}
)clear all
f(x,a,b,c) == c*(a+b*x)^3/x^2

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.003),x=-2..2,adaptive==true,unit==[1.0,1.0],
   title=="p38-2.2.30.1-3")
graph22301:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,4.0,0.003),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22302:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,6.0,0.003),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph22303:=getGraph(viewport3,1)

putGraph(viewport1,graph22302,2)
putGraph(viewport1,graph22303,3)
units(viewport1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\end{verbatim}
\[ y = \frac{c(a + bx)}{x^3} \]

\[ x^3y - bcx - ca = 0 \]

— p40-2.2.31.1-3 —

```plaintext
)clear all
f(x,a,b,c) == c*(a+b*x)/x^3

lineColorDefault(red())
viewport1:=draw(f(x,1.0,2.0,0.02),x=-4..4,adaptive==true,unit==[1.0,1.0],
    title=="p40-2.2.31.1-3")
graph22311:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,1.0,4.0,0.02),x=-4..4,adaptive==true,unit==[1.0,1.0])
graph22312:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,1.0,6.0,0.02),x=-4..4,adaptive==true,unit==[1.0,1.0])
graph22313:=getGraph(viewport3,1)

putGraph(viewport1,graph22312,2)
```
4.2. CRC GRAPHS

\[ y = \frac{c(a + bx)^2}{x^3} \]

\[ x^3y - b^2cx^2 - 2abcx - a^2c = 0 \]
\[ y = \frac{c(a + bx)^3}{x^3} \]

\[ x^3 y - b^3 cx^3 - 3ab^2 cx^2 - 3a^2 bcx - a^3 c = 0 \]
4.2. CRC GRAPHS

— p40-2.2.33.1-3 —

\( f(x, a, b, c) = c(a + b x)^3 / x^3 \)

lineColorDefault(red())
viewport1 := draw(f(x, 1.0, 2.0, 0.002), x=-2..2, adaptive=true, unit=[1.0,1.0],
    title="p40-2.2.33.1-3")
graph22331 := getGraph(viewport1,1)

lineColorDefault(green())
viewport2 := draw(f(x, 1.0, 4.0, 0.002), x=-2..2, adaptive=true, unit=[1.0,1.0])
graph22332 := getGraph(viewport2,1)

lineColorDefault(blue())
viewport3 := draw(f(x, 1.0, 6.0, 0.002), x=-2..2, adaptive=true, unit=[1.0,1.0])
graph22333 := getGraph(viewport3,1)

putGraph(viewport1, graph22332, 2)
putGraph(viewport1, graph22333, 3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeView2D(viewport1)
Functions with $a^2 + x^2$ and $x^m$

Page 42 2.3.1

\[
y = \frac{c}{(a^2 + b^2)}
\]

\[
a^2 y + x^2 y - c = 0
\]

--- p42-2.3.1.1-3 ---

```plaintext
)clear all
f(x,a,c) == c/(a^2+x^2)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.04),x=-2..2,adaptive=true,unit=[[1.0,1.0],
    title="p42-2.3.1.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,0.04),x=-2..2,adaptive=true,unit=[[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,0.04),x=-2..2,adaptive=true,unit=[[1.0,1.0])
graph3:=getGraph(viewport3,1)
```
4.2. CRC GRAPHS

\begin{verbatim}
putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\end{verbatim}

\begin{align*}
y &= \frac{cx}{(a^2 + b^2)} \\
a^2y + x^2y - cx &= 0
\end{align*}

---

\begin{verbatim}
)clear all
f(x,a,c) == c*x/(a^2+x^2)
lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.3),x=-2..2,adaptive==true,unit==[1.0,1.0],_
\end{verbatim}
\[ y = \frac{c x^2}{(a^2 + b^2)} \]
\[ a^2 y + x^2 y - cx^2 = 0 \]

--- p42-2.3.3.1-3 ---

\)
clear all
f(x,a,c) == \frac{c*x^2}{(a^2+x^2)}

lineColorDefault(red())
viewport1:=draw(f(x,0.2,1.0),x=-2..2,adaptive==true,unit=[1.0,1.0],
    title="p42-2.3.3.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,1.0),x=-2..2,adaptive==true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,1.0),x=-2..2,adaptive==true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

_____
\[ y = \frac{cx^3}{(a^2 + b^2)} \]

\[ a^2y + x^2y - cx^3 = 0 \]
4.2. CRC GRAPHS

\[ y = \frac{c}{x(a^2 + b^2)} \]

\[ a^2xy + x^3y - c = 0 \]
$y = \frac{c}{x^{2}(a^{2} + b^{2})}$

$a^{2}x^{2}y + x^{4}y - c = 0$
\[ y = cx(a^2 + x^2) \]
\[ y - a^2cx - cx^3 = 0 \]
4.2. CRC GRAPHS

```plaintext
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```

---

Page 44 2.3.8

\[
y = cx^2(a^2 + x^2)
\]

\[
y - a^2cx^2 - cx^4 = 0
\]

```plaintext
)clear all
f(x,a,c) == c*x^2*(a^2+x^2)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0],
    title="p44-2.3.8.1-3")
graph1:=getGraph(viewport1,1)
```
Functions with $a^2 - x^2$ and $x^m$

Page 46 2.4.1

$$y = \frac{c}{(a^2 - x^2)}$$

$$a^2 y - x^2 y - c = 0$$
4.2. CRC GRAPHS

— p46-2.4.1.1-3 —

)clear all
f(x,a,c) == c/(a^2-x^2)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.03),x=-2..2,adaptive==true,unit==[1.0,1.0],_
title=="p46-2.4.1.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,0.03),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,0.03),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\[ y = \frac{cx}{a^2 - x^2} \]
\[ a^2 y - x^2 y - cx = 0 \]

--- p46-2.4.2.1-3 ---

```plaintext
)clear all
f(x,a,c) == c*x/(a^2-x^2)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0],
   title="p46-2.4.2.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
```
4.2. CRC GRAPHS

\begin{align*}
\frac{cy^2}{(a^2-x^2)} & = y \\
ax^2 - x^2y - cx^2 & = 0
\end{align*}

\[ f(x,a,c) = c*x^2/(a^2-x^2) \]

\text{clear all}
\text{lineColorDefault(red())}
\text{viewport1:=draw(f(x,0.2,0.2),x=-2..2,adaptive==true,unit==[1.0,1.0],title="p46-2.4.3.1-3")}
\[ y = \frac{cx^3}{(a^2 - x^2)} \]

\[ a^2 y - x^2 y - cx^3 = 0 \]
4.2. CRC GRAPHS

--- p46-2.4.4.1-3 ---

\( f(x,a,c) = \frac{c \cdot x^3}{a^2 - x^2} \)

```plaintext
\textcolor{red}{\text{lineColorDefault(red())}}
\text{viewport1:=draw(f(x,0.2,0.2),x=-2..2,adaptive=true,unit=[1.0,1.0],
\quad title="p46-2.4.4.1-3")}
\text{graph1:=getGraph(viewport1,1)}
\textcolor{green}{\text{lineColorDefault(green())}}
\text{viewport2:=draw(f(x,0.5,0.2),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\text{graph2:=getGraph(viewport2,1)}
\textcolor{blue}{\text{lineColorDefault(blue())}}
\text{viewport3:=draw(f(x,0.8,0.2),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\text{graph3:=getGraph(viewport3,1)}
\text{putGraph(viewport1,graph2,2)}
\text{putGraph(viewport1,graph3,3)}
\text{units(viewport1,1,"on")}
\text{points(viewport1,1,"off")}
\text{points(viewport1,2,"off")}
\text{points(viewport1,3,"off")}
\text{makeViewport2D(viewport1)}
```
\[ y = \frac{c}{x(a^2 - x^2)} \]

\[ a^2 xy - x^3 y - c = 0 \]

— p48-2.4.5.1-3 —

clear all
f(x,a,c) == c/(x*(a^2-x^2))

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.001),x=-2..2,adaptive==true,unit==[1.0,1.0],
   title=="p48-2.4.5.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,0.001),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,0.001),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
\[ y = \frac{c}{x^2(a^2 - x^2)} \]
\[ a^2x^2y - x^4y - c = 0 \]
lineColorDefault(green())
viewport2:=draw(f(x,0.5,0.0003),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)
lineColorDefault(blue())
viewport3:=draw(f(x,0.8,0.0003),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)
putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

\[
y = cx(a^2 - x^2)
\]
\[
y - a^2 cx + cx^3 = 0
\]

Page 48 2.4.7

— p48-2.4.7.1-3 —
4.2. CRC GRAPHS

```plaintext
)clear all
f(x,a,c) == c*x*(a^2-x^2)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0],
  title=="p48-2.4.7.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,1.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```

---

```plaintext
p48-2.4.7.1-3
```
Page 48 2.4.8

\[ y = cx^2(a^2 - x^2) \]
\[ y - a^2 cx^2 + cx^4 = 0 \]

--- p48-2.4.8.1-3 ---

)clear all
f(x,a,c) == c*x^2*(a^2-x^2)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,4.0),x=-2..2,adaptive==true,unit==[1.0,1.0],_
           title=="p48-2.4.8.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.5,4.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,4.0),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
4.2. CRC GRAPHS

Functions with $a^3 + x^3$ and $x^m$

Page 50 2.5.1

$$y = \frac{c}{(a^3 + x^3)}$$

$$a^3y + x^3y - c = 0$$

--- p50-2.5.1.1-3 ---

```plaintext
)clear all
f(x,a,c) == c/(a^3+x^3)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.01),x=-2..2,adaptive=true,unit=[[1.0,1.0]],
       title="p50-2.5.1.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.3,0.01),x=-2..2,adaptive=true,unit=[[1.0,1.0]])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.4,0.01),x=-2..2,adaptive=true,unit=[[1.0,1.0]])
graph3:=getGraph(viewport3,1)
```
\[
y = \frac{cx}{(a^3 + x^3)}
\]

\[a^3y + x^3y - cx = 0\]
4.2. CRC GRAPHS

```
title="p50-2.5.2.1-3")
g1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.3,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0])
g2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.5,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0])
g3:=getGraph(viewport3,1)

putGraph(viewport1,g2,2)
putGraph(viewport1,g3,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,"off")
points(viewport1,"off")
makeViewport2D(viewport1)
```

\[
y = \frac{cx^2}{(a^3 + x^3)}
\]
$$a^3y + x^3y - cx^2 = 0$$

--- p50-2.5.3.1-3 ---

)clear all
f(x,a,c) == c*x^2/(a^3+x^3)

lineColorDefault(red())
viewport1:=draw(f(x,0.1,0.01),x=-2..2,adaptive==true,unit==[1.0,1.0],
title=="p50-2.5.3.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.3,0.01),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.5,0.01),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
4.2. CRC GRAPHS

\[ y = \frac{cx^3}{(a^3 + x^3)} \]

\[ a^3y + x^3y - cx^3 = 0 \]

— p50-2.5.4.1-3 —

)clear all
f(x,a,c) == c*x^3/(a^3+x^3)

lineColorDefault(red())
viewport1:=draw(f(x,0.1,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0],
title=="p50-2.5.4.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.3,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.5,0.02),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
\[ y = \frac{c}{x(a^3 + x^3)} \]

\[ a^3xy + x^4y - c = 0 \]

--- p50-2.5.5.1-3 ---
y = cx(a^3 + x^3)

y - a^3 cx - cx^4 = 0
132 CHAPTER 4. CRC STANDARD CURVES AND SURFACES [7]

)clear all
f(x,a,c) == c*x*(a^3+x^3)

lineColorDefault(red())
viewport1:=draw(f(x,0.5,1.0),x=-2..2,adaptive==true,unit=[[1.0,1.0],
    title="p50-2.5.6.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.7,1.0),x=-2..2,adaptive==true,unit=[[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.9,1.0),x=-2..2,adaptive==true,unit=[[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
Functions with $a^3 - x^3$ and $x^m$

Page 52 2.6.1

$$y = \frac{c}{(a^3 - x^3)}$$

$$a^3y - x^3y - c = 0$$

— p52-2.6.1.1-3 —

)clear all
f(x,a,c) == c/(a^3-x^3)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0],
    title="p52-2.6.1.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.3,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.4,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\[ y = \frac{cx}{a^3 - x^3} \]
\[ a^3y - x^3y - cx = 0 \]
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

\[ y = \frac{cx^2}{(a^3 - x^3)} \]
\[ a^3 y - x^3 y - cx^2 = 0 \]

)clear all
f(x,a,c) == c*x^2/(a^3-x^3)
lineColorDefault(red())
viewport1:=draw(f(x,0.1,0.1),x=-2..2,adaptive==true,unit==[1.0,1.0],
               title="p52-2.6.3.1-3")
\[ y = \frac{cx^3}{(a^3 - x^3)} \]
\[ a^3y - x^3y - cx^3 = 0 \]
4.2. CRC GRAPHS

--- p52-2.6.4.1-3 ---

)clear all
f(x,a,c) == c*x^3/(a^3-x^3)

lineColorDefault(red())
viewport1:=draw(f(x,0.1,0.2),x=-2..2,adaptive=true,unit=[1.0,1.0],
              title="p52-2.6.4.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.3,0.2),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.5,0.2),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\[ y = \frac{c}{x(a^3 - x^3)} \]

\[ a^3xy - x^4y - c = 0 \]

--- p52-2.6.5.1-3 ---

```plaintext
)clear all
f(x,a,c) == c/(x*(a^3-x^3))

lineColorDefault(red())
viewport1:=draw(f(x,0.5,0.01),x=-2..2,adaptive==true,unit==[1.0,1.0],
        title=="p52-2.6.5.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.7,0.01),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.9,0.01),x=-2..2,adaptive==true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
```
4.2. CRC GRAPHS

```plaintext
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```

\[ y = c x (a^3 - x^3) \]
\[ y - a^3 c x + c x^4 = 0 \]
Functions with $a^4 + x^4$ and $x^m$

Page 54 2.7.1

$$y = \frac{c}{(a^4 + x^4)}$$
4.2. CRC GRAPHS

\[ a^4 y + x^4 y - c = 0 \]

--- p54-2.7.1.1-3 ---

\)
clear all
f(x,a,c) \equiv \frac{c}{a^4+x^4}
clineColorDefault(red())
viewport1:=draw(f(x,0.3,0.007),x=-2..2,adaptive=true,unit=[1.0,1.0],
  title="p54-2.7.1.1-3")
graph1:=getGraph(viewport1,1)
clineColorDefault(green())
viewport2:=draw(f(x,0.4,0.007),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)
clineColorDefault(blue())
viewport3:=draw(f(x,0.5,0.007),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)
putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

---
y = \frac{cx}{(a^4 + x^4)}

a^4y + x^4y - cx = 0

— p54-2.7.2.1-3 —
Page 54 2.7.3

\[ y = \frac{c x^2}{(a^4 + x^4)} \]

\[ a^4 y + x^4 y - c x^2 = 0 \]
\[
y = \frac{c x^3}{(a^4 + x^4)}
\]

\[a^4 y + x^4 y - cx^3 = 0\]
4.2. CRC GRAPHS

--- p54-2.7.4.1-3 ---

)clear all
f(x,a,c) == c*x^3/(a^4+x^4)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.25),x=-2..2,adaptive=true,unit==[1.0,1.0], _
title=="p54-2.7.4.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.4,0.25),x=-2..2,adaptive=true,unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.6,0.25),x=-2..2,adaptive=true,unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
\[ y = \frac{cx^4}{(a^4 + x^4)} \]

\[ a^4 y + x^4 y - cx^4 = 0 \]
4.2. CRC GRAPHS

\[
y = cx(a^4 + x^4)
\]
\[
y - a^4cx - cx^5 = 0
\]
Functions with $a^4 - x^4$ and $x^m$

Page 56 2.8.1

$$y = \frac{c}{(a^4 - x^4)}$$
\[ a^4 y - x^4 y - c = 0 \]

--- p56-2.8.1.1-3 ---

```plaintext
)clear all
f(x,a,c) == c/(a^4-x^4)

lineColorDefault(red())
viewport1:=draw(f(x,0.4,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0],
    title="p56-2.8.1.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.6,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.8,0.01),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```
\[ y = \frac{cx}{(a^4 - x^4)} \]

\[ a^4y - x^4y - cx = 0 \]
4.2. CRC GRAPHS

```plaintext
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```

---

Page 56 2.8.3

\[ y = \frac{cx^2}{(a^4 - x^4)} \]

\[ a^4 y - x^4 y - c x^2 = 0 \]

```plaintext
)clear all
f(x,a,c) == c*x^2/(a^4-x^4)

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0],
title="p56-2.8.3.1-3")
```
\[ y = \frac{cx^3}{(a^4 - x^4)} \]
\[ a^4y - x^4y - cx^3 = 0 \]
4.2. CRC GRAPHS

— p56-2.8.4.1-3 —

\[ f(x,a,c) = \frac{c \cdot x^3}{a^4 - x^4} \]

lineColorDefault(red())
viewport1:=draw(f(x,0.2,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0],
    title="p56-2.8.4.1-3")
graph1:=getGraph(viewport1,1)

lineColorDefault(green())
viewport2:=draw(f(x,0.4,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph2:=getGraph(viewport2,1)

lineColorDefault(blue())
viewport3:=draw(f(x,0.6,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

———
\[ y = \frac{cx^4}{(a^4 - x^4)} \]

\[ a^4y - x^4y - cx^4 = 0 \]

--- p56-2.8.5.1-3 ---

```latex
\texttt{\textasciitilde clear all}
\texttt{f(x,a,c) == c*x^4/(a^4-x^4)}
\texttt{lineColorDefault(red())}
\texttt{viewport1:=draw(f(x,0.2,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0],
   title="p56-2.8.5.1-3")}
\texttt{graph1:=getGraph(viewport1,1)}
\texttt{lineColorDefault(green())}
\texttt{viewport2:=draw(f(x,0.5,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\texttt{graph2:=getGraph(viewport2,1)}
\texttt{lineColorDefault(blue())}
\texttt{viewport3:=draw(f(x,0.8,0.1),x=-2..2,adaptive=true,unit=[1.0,1.0])}
\texttt{graph3:=getGraph(viewport3,1)}
\texttt{putGraph(viewport1,graph2,2)}
```
4.2. CRC GRAPHS

\[
y = cx(a^4 - x^4)
\]

\[
y - a^4 cx + cx^5 = 0
\]

--- p56-2.8.6.1-3 ---

```plaintext
clear all
f(x,a,c) == c*x*(a^4-x^4)

lineColorDefault(red())
viewport1:=draw(f(x,0.4,1.0),x=-2..2,adaptive=true,unit=[1.0,1.0],
    title="p56-2.8.6.1-3")
graph1:=getGraph(viewport1,1)
```
Functions with $(a + bx)^{1/2}$ and $x^m$

Page 58 2.9.1

Parabola

$$y = c(a + bx)^{1/2}$$
4.2. CRC GRAPHS

\[ y^2 - bc^2 x - ac^2 = 0 \]

--- p58-2.9.1.1-3 ---

\)
clear all
f1(x,y) == y^2 - x/8 - 1/2

lineColorDefault(red())
viewport1:=drawer(f1(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
        unit==[1.0,1.0],title=="p58-2.9.1.1-3")
graph1:=getGraph(viewport1,1)

f2(x,y) == y^2 - x - 1/2

lineColorDefault(green())
viewport2:=drawer(f2(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
        unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

f3(x,y) == y^2 - 2*x - 1/2

lineColorDefault(blue())
viewport3:=drawer(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
        unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
Trisectrix of Catalan... fails "singular pts in region of sketch"

\[ y = cx(a + bx)^{1/2} \]

\[ y^2 - bc^2x^3 - ac^2x^2 = 0 \]
4.2. CRC GRAPHS

```plaintext
lineColorDefault(blue())
viewport3:=draw(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)
putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
```

Page 58 2.9.3

fails with "singular pts in region of sketch"

\[ y = cx^2(a + bx)^{1/2} \]
\[ y^2 - bc^2x^5 - ac^2x^4 = 0 \]

— p58-2.9.3.1-3 —

```plaintext
)clear all
f1(x,y) == y^2 - x^5/2 - x^4/2
lineColorDefault(red())
viewport1:=draw(f1(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0],title="p58-2.9.3.1-3")
graph1:=getGraph(viewport1,1)
f2(x,y) == y^2 - x^5 - x^4/2
lineColorDefault(green())
viewport2:=draw(f2(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)
f3(x,y) == y^2 - 2*x^5 - x^4/2
lineColorDefault(blue())
viewport3:=draw(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)
```
Page 58 2.9.4

\[ y = \sqrt{c(a + bx)} \frac{1}{x} \]
\[ x^2y^2 - c^2bx - c^2a = 0 \]

--- p58-2.9.4.1-3 ---
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

\[
y = c(a + bx)^{1/2}/x^2
\]
\[
x^4 y^2 - c^2 bx - c^2 a = 0
\]

\—— p58-2.9.5.1-3 ——

)clear all
f1(x,y) == x^2*y^2 - x/200 - 1/200

lineColorDefault(red())
viewport1:=draw(f1(x,y)=0,x,y,range==[-2.0..2.0,-2.0..2.0],adaptive==true,_
unit==[1.0,1.0],title=="p58-2.9.5.1-3")
graph1:=getGraph(viewport1,1)
f2(x,y) == x^2*y^2 - x/100 - 1/200
\[ y = \frac{c}{(a + bx)^{1/2}} \]

\[
ay^2 + bxy^2 - c^2 = 0
\]
4.2. CRC GRAPHS

— p58-2.9.6.1-3 —

\[
\text{clear all}
\]
\[
f1(x,y) = (x + 1)y^2 - 1/4
\]
\[
\text{lineColorDefault(red())}
\]
\[
\text{viewport1:=draw(f1(x,y)=0,x,y,range=[-4.0..4.0,-4.0..4.0],adaptive=true,}
\]
\[
\text{unit=[1.0,1.0],title="p58-2.9.6.1-3")}
\]
\[
\text{graph1:=getGraph(viewport1,1)}
\]
\[
f2(x,y) = (2x+1)y^2 - 1/4
\]
\[
\text{lineColorDefault(green())}
\]
\[
\text{viewport2:=draw(f2(x,y)=0,x,y,range=[-4.0..4.0,-4.0..4.0],adaptive=true,}
\]
\[
\text{unit=[1.0,1.0])}
\]
\[
\text{graph2:=getGraph(viewport2,1)}
\]
\[
f3(x,y) = (4x+1)y^2 - 1/4
\]
\[
\text{lineColorDefault(blue())}
\]
\[
\text{viewport3:=draw(f3(x,y)=0,x,y,range=[-4.0..4.0,-4.0..4.0],adaptive=true,}
\]
\[
\text{unit=[1.0,1.0])}
\]
\[
\text{graph3:=getGraph(viewport3,1)}
\]
\[
\text{putGraph(viewport1,graph2,2)}
\]
\[
\text{putGraph(viewport1,graph3,3)}
\]
\[
\text{units(viewport1,1,"on")}
\]
\[
\text{points(viewport1,1,"off")}
\]
\[
\text{points(viewport1,2,"off")}
\]
\[
\text{points(viewport1,3,"off")}
\]
\[
\text{makeViewport2D(viewport1)}
\]
singular pts in region of sketch

\[ y = \frac{cx}{(a + bx)^{1/2}} \]

\[ ay^2 + bxy^2 - c^2x^2 = 0 \]

--- p60-2.9.7.1-3 ---
4.2. CRC GRAPHS

lineColorDefault(blue())
viewport3:=draw(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)
putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

---

Page 60 2.9.8

singular pts in region of sketch

\[
y = \frac{cx^2}{(a + bx)^{1/2}}
\]

\[
ay^2 + bxy^2 - c^2x^4 = 0
\]

— p60-2.9.8.1-3 —

)clear all
f1(x,y) == (x + 1)*y^2 - x^4

lineColorDefault(red())
viewport1:=draw(f1(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0],title="p60-2.9.8.1-3")
graph1:=getGraph(viewport1,1)

f2(x,y) == (2*x+1)*y^2 - x^4

lineColorDefault(green())
viewport2:=draw(f2(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

f3(x,y) == (4*x+1)*y^2 - x^4

lineColorDefault(blue())
viewport3:=draw(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)
Page 60 2.9.9

\[
y = \frac{c}{x(a + bx)^{1/2}}
\]

\[ax^2y^2 + bx^3y^2 - c^2 = 0\]

— p60-2.9.9.1-3 —

)clear all
f1(x,y) == (4/5*x^3 + x^2)*y*2 - 1/25

lineColorDefault(red())
viewport1:=draw(f1(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0],title=="p60-2.9.9.1-3")
graph1:=getGraph(viewport1,1)

f2(x,y) == (x^3 + x^2)*y^2 - 1/25

lineColorDefault(green())
viewport2:=draw(f2(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

f3(x,y) == (6/5*x^3 + x^2)*y^2 - 1/25

lineColorDefault(blue())
viewport3:=draw(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
4.2. CRC GRAPHS

points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)

\[ y = \frac{c}{x^2(a + bx)^{1/2}} \]
\[ ax^4y^2 + bx^5y^2 - c^2 = 0 \]

-- p60-2.9.10.1-3 --

)clear all
f1(x,y) == (4/5*x^5 + x^4)*y^2 - 1/100

lineColorDefault(red())
viewport1:=draw(f1(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0],title=="p60-2.9.10.1-3")
graph1:=getGraph(viewport1,1)

f2(x,y) == (x^5 + x^4)*y^2 - 1/100
lineColorDefault(green())
viewport2:=draw(f2(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

f3(x,y) == (6/5*x^5 + x^4)*y^2 - 1/100
lineColorDefault(blue())
viewport3:=draw(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
4.2. CRC GRAPHS

\[ y = \frac{cx^{1/2}}{(a + bx)^{1/2}} \]

\[ y^2 - ac^2x - bc^2x^2 = 0 \]

--- p60-2.9.11.1-6 ---

```plaintext
)clear all
f1(x,y) == y^2 + 2*x^2 - 2*x
lineColorDefault(color(1))
viewport1:=draw(f1(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0],title=="p60-2.9.11.1-6")
graph1:=getGraph(viewport1,1)

f2(x,y) == y^2 + 3*x^2 - 2*x
lineColorDefault(color(2))
viewport2:=draw(f2(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0])
graph2:=getGraph(viewport2,1)

f3(x,y) == y^2 + 4*x^2 - 2*x
lineColorDefault(color(3))
viewport3:=draw(f3(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0])
graph3:=getGraph(viewport3,1)

f4(x,y) == y^2 - x^2 - 3*x/10
lineColorDefault(color(4))
viewport4:=draw(f4(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0])
graph4:=getGraph(viewport4,1)

f5(x,y) == y^2 - x^2 - x/2
lineColorDefault(color(5))
viewport5:=draw(f5(x,y)=0,x,y,range==[-4.0..4.0,-4.0..4.0],adaptive==true,_
    unit==[1.0,1.0])
graph5:=getGraph(viewport5,1)

f6(x,y) == y^2 - x^2 - 7*x/10
lineColorDefault(color(6))
```
viewport6:=draw(f6(x,y)=0,x,y,range=[-4.0..4.0,-4.0..4.0],adaptive=true, unit=[1.0,1.0])
graph6:=getGraph(viewport6,1)
putGraph(viewport1,graph2,2)
putGraph(viewport1,graph3,3)
putGraph(viewport1,graph4,4)
putGraph(viewport1,graph5,5)
putGraph(viewport1,graph6,6)
units(viewport1,1,"on")
points(viewport1,1,"off")
points(viewport1,2,"off")
points(viewport1,3,"off")
makeViewport2D(viewport1)
Chapter 5

Pasta by Design[4]

This is a book that combines a taxonomy of pasta shapes with the Mathematica equations that realize those shapes in three dimensions. We implemented examples from this book as a graphics test suite for Axiom.
5.1 Acini Di Pepe

The smallest member of the *postine minute* (tiny pasta) family, *acini de pepe* (peppercorns) are most suited to consommés (clear soups), with the occasional addition of croutons and diced greens. Made of durum wheat flour and eggs, acini di pepe are commonly used in the Italian-American “wedding soup”, a broth of vegetables and meat.
These shell-like ravioli from Piedmont, northern Italy, are fashioned from small pieces of flattened dough made of wheat flour and egg, and are often filled with braised veal, port, vegetables or cheese. The true agnolotto should feature a crinkled edge, cut using a fluted pasta wheel. Recommended with melted butter and sage.
5.3 Anellini

The diminutive onellini (small rings) are part of the extended postine minute (tiny pasta) clan. Their thickness varies between only 1.15 and 1.20 mm, and they are therefore usually found in light soups together with croutons and thinly sliced vegetables. This pasta may also be found served in a timballo (baked pasta dish).
5.4. BUCATINI

5.4 Bucatini

--- Bucatini ---

\[
\begin{align*}
X(i,j) &= 0.3 \cos(i/30\pi) \\
Y(i,j) &= 0.3 \sin(i/30\pi) \\
Z(i,j) &= j/45
\end{align*}
\]

\begin{verbatim}
canvas := createThreeSpace() 
cf(x:DFLOAT,y:DFLOAT):DFLOAT == 1.0 
makeObject(surface(X(i,j),Y(i,j),Z(i,j)),i=0..60,j=0..90,space==canvas, _
    colorFunction==cf,style=="smooth") 
makeObject(surface(X(i,j)/2,Y(i,j)/2,Z(i,j)),i=0..60,j=0..90,space==canvas, _
    colorFunction==cf,style=="smooth") 
vp:=makeViewport3D(canvas,style=="smooth",title=="Bucatini") 
colorDef(vp,yellow(),yellow()) 
axes(vp,"off") 
zoom(vp,2.0,2.0,2.0)
\end{verbatim}

---

Bucatini (pierced) pasta is commonly served as a pastasciutta (pasta boiled, drained, and dished up with a sauce, rather than in broth). Its best known accompaniment is amatriciana: a hearty traditional sauce made with dried port, Pecorino Romano and tomato sauce, and named after the medieval town of Amatrice in central Italy.
5.5 Buccoli

--- Buccoli ---

\[ X(i,j) = (0.7 + 0.2\sin(21\pi j/250)\pi)\cos(i/20\pi) \]
\[ Y(i,j) = (0.7 + 0.2\sin(21\pi j/250)\pi)\sin(i/20\pi) \]
\[ Z(i,j) = 39.0i/1000. + 1.5\sin(j/50\pi) \]
\[ cf(x:\text{DFLOAT}, y:\text{DFLOAT}):\text{DFLOAT} = 1.0 \]
\[ v3d := \text{draw(surface}(X(i,j),Y(i,j),Z(i,j)),i=0..200,j=0..25, \]
\[ \quad \text{style}=="smooth", \text{title}"\text{Buccoli}"\text{.colorFunction}==cf \]
\[ \text{colorDef}(v3d,\text{yellow()},\text{yellow}()) \]
\[ \text{axes}(v3d,"off") \]
\[ \text{zoom}(v3d,2.0,2.0,2.0) \]

---

A spiral-shaped example from the pasta corta (short pasta) family, and of rather uncertain pedigree, buccioli are suitable in a mushroom and sausage dish. They are also excellent with a tomato aubergine, pesto, and ricotta salad.
5.6. CALAMARETTI

5.6 Calamaretti

Calamaretti

Literally “little squids”, calamaretti are small ring-shaped pasta cooked as pastasciutta (pasta boiled and drained) then dished up with a tomato-, egg-, or cheese-based sauce. Their shape means that calamaretti hold both chunky and thin sauces equally well. Fittingly, they are often served with seafood.
5.7 Cannelloni

--- Cannelloni ---

\[
\begin{align*}
X(i,j) &= (1+j/100)\cos(i\pi/55) + 0.5\cos(j\pi/100) + 0.1\cos(i\pi/55 + j\pi/125) \\
Y(i,j) &= 1.3\sin(i\pi/55) + 0.3\sin(j\pi/100) \\
Z(i,j) &= 7.0j/50.
\end{align*}
\]

\[\text{cf}(x:\text{DFLOAT},y:\text{DFLOAT}):\text{DFLOAT} \rightarrow 1.0\]

v3d := draw(surface(X(i,j),Y(i,j),Z(i,j)),i=0..110,j=0..50,
          style=="smooth",title=="Cannelloni",colorFunction==cf)

colorDef(v3d,yellow(),yellow())
axes(v3d,"off")
zoom(v3d,2.0,2.0,2.0)

---

Made with wheat flour, eggs, and olive oil, cannelloni (big tubes) originate as strips of pasta shaped into perfect cylinders, which can be stuffed with meat, vegetables, or ricotta. The stuffed cannelloni are covered with a creamy besciamella sauce, a sprinkling of Parmigiano-Reggiano cheese and then oven-baked.
5.8 Cannolicchi Rigati

Known as “little tubes”, cannolicchi exist both in a rigati (grooved) and lisci (smooth) form. These hollow pasta corta (short pasta) come in various diameters and are often served with seafood. Cannolicchi hail from Campania in southern Italy.
### 5.9 Capellini

An extra-fine rod-like pasta *capellini* (thin hair) may be served in a light broth, but also combine perfectly with butter, nutmeg, or lemon. This variety (or its even more slender relative, *capelli d’angelo* (angel hair) is sometimes used to form the basis of an unusual sweet pasta dish, made with lemons and almonds, called *torta ricciolina*.
5.10. CAPPHELLETTI

5.10 Cappelletti

This pasta is customarily served as the first course of a traditional north Italian Christmas meal, dished up in a chicken brodo (broth). Typically, it is the children of a household who prepare the cappelletti (little hats) on Christmas Eve, filling the pasta parcels (made from wheat flour and fresh eggs) with mixed meats or soft cheeses, such as ricotta.
5.11 Casarecce

Easily identified by their unique s-shaped cross-section casarecce (home-made) are best cooked as postasciutta (pasta boiled, drained and dished up with a sauce). Often casarecce are served with a classic ragù and topped with a sprinkle of pepper and Parmigiano-Reggiano cheese.
The manufacturer Bailla has recently created this elegant pasta shape. According to its maker, they were originally called *paguri* (hermit crabs) but renamed *castellane* (castle dwellers). The sturdy form and rich nutty taste of *castellane* stand up to hearty meals and full-flavoured sauces.
5.13 Cavatappi

Perfect with chunky sauces made from lamb or pork, cavatappi (corkscrews) are 36 mm-long, hollow helicoidal tubes. As well as an accompaniment to creamy sauces, such as boscaiola (woodsman’s) sauce, they are also often used in oven-baked cheese-topped dishes, or in salads with pesto.
5.14 Cavatelli

Popular in the south of Italy, and related in shape to the longer twisted casareccia, cavatelli can be served alla puttanesca (with a sauce containing chilli, garlic, capers, and anchovies). They can also be added to a salad with olive oil, sautéed crushed garlic and a dusting of soft cheese.
5.15 Chifferi Rigati

This pasta - available in both rigoti (grooved) and lisci (smooth) forms - is typically cooked in broth or served in rogu alla bolognese, though chifferi rigati also make an excellent addition to salads with carrot, red pepper and courgette. Chifferi rigoti bear a resemblance to, and the term is a transliteration of, the Austrian 'kipfel' sweet.
5.16 Colonne Pompeii

\[
\begin{align*}
X_0(i,j) &= \_ \\
&\quad \text{if (} j \leq 50 \text{)} \_ \\
&\quad \quad \text{then } 2\cos(i\pi/20) \_ \\
&\quad \quad \quad \text{else } 2\cos(i\pi/20)\cos(-j\pi/25) \\
Y_0(i,j) &= \_ \\
&\quad \text{if (} j \leq 50 \text{)} \_ \\
&\quad \quad \text{then } 0.0 \_ \\
&\quad \quad \quad \text{else } 2\cos(i\pi/20)\sin(-j\pi/25) + 3\sin((j-50)\pi/200) \\
Z_0(i,j) &= \_ \\
&\quad \text{if (} j \leq 50 \text{)} \_ \\
&\quad \quad \text{then } \sin(i\pi/20)+12 \_ \\
&\quad \quad \quad \text{else } \sin(i\pi/20)+6.0*j/25.0 \\
X_1(i,j) &= \_ \\
&\quad \text{if (} j \leq 200 \text{)} \_ \\
&\quad \quad \text{then } 2\cos(i\pi/20)\cos(-j\pi/25+2\pi/3) \_ \\
&\quad \quad \quad \text{else } 2\cos(i\pi/20)\sin(-28\pi/3) \\
Y_1(i,j) &= \_ \\
&\quad \text{if (} j \leq 200 \text{)} \_ \\
&\quad \quad \text{then } 2\cos(i\pi/20)\sin(-j\pi/20 + 2\pi/3) + 3\sin(j\pi/200) \_ \\
&\quad \quad \quad \text{else } 2\cos(i\pi/20)\sin(-28\pi/3) \\
Z_1(i,j) &= \_ \\
&\quad \text{if (} j \leq 200 \text{)} \_ \\
&\quad \quad \text{then } 12+\sin(i\pi/20)+6.0*j/25.0 \_ \\
&\quad \quad \quad \text{else } \sin(i\pi/20)+60 \\
X_2(i,j) &= \_ \\
&\quad \text{if (} j \leq 200 \text{)} \_ \\
\end{align*}
\]
Then $2\cos(i\pi/20)\cos(-j\pi/25+4\pi/3)$ \else $2\cos(i\pi/20)\sin(-28\pi/3)$

$Y2(i,j) = _$
  \begin{align*}
    \text{if } (j \leq 200) \text{ then } & 2\cos(i\pi/200)\sin(-j\pi/25+4\pi/3)+3\sin(j\pi/200) \text{ else } & 2\cos(i\pi/20)\sin(-28\pi/3) \\
  \end{align*}$

vsp:=createThreeSpace()
makeObject(surface(X0(i,j),Y0(i,j),Z0(i,j)),i=0..10,j=0..250,space==vsp)
makeObject(surface(X1(i,j),Y1(i,j),Z1(i,j)),i=0..10,j=0..250,space==vsp)
makeObject(surface(X2(i,j),Y2(i,j),Z1(i,j)),i=0..10,j=0..250,space==vsp)
vp:=makeViewport3D(vsp,title=="Colonne Pompeii",style=="smooth")
colorDef(vp,yellow(),yellow())
axes(vp,"off")
zoom(vp,3.0,3.0,3.0)

This ornate pasta (originally from Campania, southern Italy) is similar in shape to fusilloni (a large fusilli) but is substantially longer. Colonne Pompeii (columns of Pompeii) are best served with a seasoning of fresh basil, pine nuts, finely sliced garlic and olive oil, topped with a sprinkling of freshly grated Parmigiano-Reggiano.
5.17 Conchiglie Rigate

Shaped like their namesake, *conchiglie* (shells) exist in both *rigate* (grooved) and *lisce* (smooth) forms. Suited to light tomato sauces, ricotta cheese or *pesto genovese*, *conchiglie* hold flavourings in their grooves and cunningly designed shell. Smaller versions are used in soups, while larger shells are more commonly served with a sauce.
5.18 Conchigliette Lisce

Typically found in central and southern Italy (notably Campania), conchigliette lisce (small smooth shells) can be served in soups such as minestrone. Alternatively, these shells can accompany a meat- or vegetable-based sauce.
### 5.19 Conchiglioni Rigate

---

**Conchiglioni Rigate**

- **Conchiglioni Rigate**

---

\[
\begin{align*}
\mathbf{A}(i,j) &= 0.25 \sin(j \cdot \pi/200) \cos((j+4) \cdot \pi/4) \\
\mathbf{B}(i,j) &= i/40.0 \cdot (0.1 + 0.1 \sin(j \cdot \pi/200)^6) \cdot \pi \\
\mathbf{C}(i,j) &= 2.5 \cos(j \cdot \pi/100) + 3 \sin((40-i) \cdot \pi/80)^{10} \cdot \\
& \quad \sin(j \cdot \pi/200)^{10} \sin((j-150) \cdot \pi/100) \\
\mathbf{X}(i,j) &= \mathbf{A}(i,j) + (10 + 30 \sin(j \cdot \pi/200)) \cdot \\
& \quad \sin((40.0-i)/40 \cdot (0.3 + \sin(j \cdot \pi/200)^{3} \cdot \pi) \cdot \\
& \quad \sin(\mathbf{B}(i,j)) \cos(j \cdot \pi/100) \\
\mathbf{Y}(i,j) &= \mathbf{A}(i,j) + (10 + 30 \sin(j \cdot \pi/200)) \cdot \\
& \quad \cos((40.0-i)/40 \cdot (0.3 + \sin(j \cdot \pi/200)^{3} \cdot \pi) \cdot \\
& \quad \sin(\mathbf{B}(i,j)) + \mathbf{C}(i,j) \\
\mathbf{Z}(i,j) &= 30.0 \cos(j \cdot \pi/200) \\
\end{align*}
\]

\[
\text{v3d:=draw(surface}(\mathbf{X}(i,j),\mathbf{Y}(i,j),\mathbf{Z}(i,j)),\text{i=0..40,j=0..200,} \\
\text{style}="\text{smooth},"\text{title}="\text{Conchiglioni Rigati}" \\
\text{colorDef(v3d,yellow(),yellow())} \\
\text{axes(v3d,"off")} \\
\text{rotate(v3d,45,45)} \\
\text{zoom(v3d,1.5,1.5,1.5)}
\]

---

The shape of *conchiglioni rigate* (large ribbed shells) is ideal for holding sauces and fillings (either fish or meat based) and the pasta can be baked in the oven, or placed under a grill and cooked as a gratin. *Conchiglioni rigati* are often served in the Italian-American dish *pasta primavera* (pasta in spring sauce) alongside crisp spring vegetables.
5.20 Corallini Lisci

Members of the postine minute (tiny pasta) group, corallini lisci (small smooth coral) are so called because their pierced appearance resembles the coral beads worn as jewelry in Italy. Their small size (no larger than 3.5 mm in diameter) means that corallini are best cooked in broths, such as Tuscan white bean soup.
5.21. Creste Di Galli

Part of the *pasta ripiena* (filled pasta) family, *creste di galli* (coxcombs) are identical to *galletti* except for the crest, which is smooth rather than crimped. They may be stuffed, cooked and served in a simple *marinara* (mariner’s) sauce, which contains tomato, garlic, and basil.
5.22 Coureatti

A romantically shaped scion of the postine minute (tiny pasta) clan, cuoretti (tiny hearts) are minuscule. In fact, along with acini di pepe, they are one of the smallest forms of pasta. Like all postine they may be served in soup, such as cream of chicken.
Another speciality of the Campania region of southern Italy, *ditali rigati* (grooved thimbles) are compact and typically less than 10 mm long. Like other *pastine*, they are usually found in soups such as *pasta e patate*. Their stocky shape makes them a sustaining winter snack, as well as an excellent addition to salads.
A notable member of the pasta ripiena (filled pasta) family, fagottini (little purses) are made from circles of durum wheat dough. A spoonful of ricotta, steamed vegetables or even stewed fruit is placed on the dough, and the corners are then pinched together to form a bundle. These packed dumplings are similar to ravioli, only larger.
5.25 Farfalle

Farfalle

--- Farfalle ---

\[
\begin{align*}
A(i) & = \sin((7i+16)\pi/40) \\
B(i,j) & = (7.0i/16.0)+4\sin(i\pi/80)\sin((j-10)\pi/120) \\
C(i,j) & = 10\cos((i+80)\pi/80)\sin((j+110)\pi/100)^9 \\
D(i,j) & = (7.0i/16.0)-4\sin(i\pi/80)-A(i)\sin((10-j)\pi/20)
\end{align*}
\]

--- E(i,j) was never defined. We guess at a likely function - close but wrong ---

\[
E(i,j) = _
\]

\[
\begin{align*}
& \text{if } (20 \leq i \leq 60) \\
& \quad \text{then } 7\sin((i+40)\pi/40)^3\sin((2j\pi)/10+1.1\pi)^9 \\
& \quad \text{else } C(i,j)
\end{align*}
\]

\[
F(i) = _
\]

\[
\begin{align*}
& \text{if } (20 \leq i \leq 60) \\
& \quad \text{then } 7\sin((i+40)\pi/40)^3\sin((j+110)\pi/100)^9 \\
& \quad \text{else } C(i,j)
\end{align*}
\]

\[
X(i,j) = (3.0i)/8.0+F(i)
\]

\[
Y(i,j) = _
\]

\[
\begin{align*}
& \text{if } (10 \leq j \leq 70) \\
& \quad \text{then } B(i,j)-4\sin(i\pi/80)\sin((70-j)\pi/120) \\
& \quad \text{else if } (j < 10) \\
& \quad \quad \text{then } D(i,j) \\
& \quad \quad \text{else } E(i,j)
\end{align*}
\]

\[
Z(i,j) = 3\sin((i+10)\pi/20)\sin(j\pi/80)^1.5
\]

\[
v3d:=\text{draw(surface}(X(i,j),Y(i,j),Z(i,j)),i=0..80,j=0..80,_,
\]

\[
\text{style}="\text{smooth}"\text{,title="Farfalle"})
\]

\[
\text{colorDef}(v3d,\text{yellow}(),\text{yellow}())
\]

\[
\text{axes}(v3d,"\text{off}")
\]

\[
\text{zoom}(v3d,0.5,0.5,0.5)
\]
A mixture of durum-wheat flour, eggs, and water, farfalle (butterflies) come from the Emilia-Romagna and Lombardy regions of northern Italy. They are best served in a rich carbonara sauce (made with cream, eggs, and bacon). Depending on se, farfalle might be accompanied by green peas and chicken or ham.
The small size of this well-known member of the postine minute (tiny pasta) lineage means that farfalline (tiny butterflies), are suitable for light soups, such as pomodori e robiolo (a mixture of tomato and soft cheese). A crimped pasta cutter and a central pinch create the iconic shape.
5.27 Farfalloni

---

Farfalloni

---

A(i,j) == 10*cos((i+70)*%pi/70)*sin((2*j)*%pi/175+1.1*%pi)^9
B(j) == 0.3*sin((6-j)*%pi/7+0.4*%pi)
C(i,j) == 
  if ((17 <= i) and (i <= 52)) 
    then 7*sin((i+35)*%pi/35)^3*sin((2*j*%pi)/175+1.1*%pi)^9 
  else A(i,j)
D(i,j) == (j/2.0)+4*sin(i*%pi/70)*sin((j-10)*%pi/100) - 
  4*sin(i*%pi/70)*sin((60-j)*%pi/100)
E(i,j) == (j/2.0)+4*sin(i*%pi/70)+0.3*sin((2*i+2.8)*%pi/7)*sin((j-60)*%pi/20)
F(i,j) == (j/2.0)-4*sin(i*%pi/70)-0.3*sin((2*i+2.8)*%pi/7)*sin((10-j)*%pi/20)
X(i,j) == (3.0*i)/7.0+C(i,j)
Y(i,j) == 
  if ((10 <= j) and (j <= 60)) 
    then D(i,j) 
  else if (j <= 10) 
    then F(i,j) 
  else E(i,j)
Z(i,j) == 3*sin((2*i+17.5)*%pi/35.)*sin(j*%pi/70)^1.5
v3d:=draw(surface(X(i,j),Y(i,j),Z(i,j)),i=0..70,j=0..70, 
  style="smooth",title="Farfalloni")
colorDef(v3d,yellow(),yellow())
axes(v3d,"off")
zoom(v3d,2.5,2.5,2.5,2.5)
5.27. FARFALLONI

Like farfalle, farfalloni (large butterflies) are well matched by a tomato- or butter-based sauce with peas and ham. They are also perfect with marrow vegetables such as roast courgette or pureed pumpkin, topped with Parmigiano-Reggiano and a sprinkling of noce moscata (nutmeg).
5.28 Festonati

This smooth member of the pasta corta (short pasta) family is named after 'festoons' (decorative lengths of fabric with the rippled profile of a garland). Festonati can be served with grilled aubergine or home-grown tomatoes, topped with grated scamorza, fresh basil, olive oil, garlic, and red chilli flakes.
5.29  Fettuccine

---  Fettuccine  ---

\[
\begin{align*}
X(i,j) &= 1.8 \sin((4i) \pi/375) \\
Y(i,y) &= 1.6 \cos((6i) \pi/375) \sin((3i) \pi/750) \\
Z(i,j) &= i/75.0 + j/20.0
\end{align*}
\]

\[
v3d:=\text{draw} (\text{surface}(X(i,j),Y(i,j),Z(i,j)),i=0..150,j=0..10, _ \\
\text{style}=="\text{smooth},\text{title}"=="\text{Fettuccine}" \\
\text{colorDef}(v3d,\text{yellow()},\text{yellow}()) \\
\text{axes}(v3d,"\text{off}")
\]

This famous pasta lunga (long pasta) is made with durum-wheat flour, water, and in the case of fettuccine all'ovo, eggs ideally within days of laying. Fettuccine (little ribbons) hail from the Lazio region. Popular in many dishes, they are an ideal accompaniment to Alfredo sauce, a rich mix of cream, parmesan, garlic, and parsley.
5.30  Fiocchi Rigati

A distant relative of the farfalle family, fiocchi rigati (grooved flakes) are smaller than either farfalloni or farfalle, but larger than farfalline. Their corrugated surface collects more sauce than a typical farfalle. For a more unusual disk, fiocchi rigati can be served in a tomato and vodka sauce.
5.31  Fisarmoniche

— Fisarmoniche —

\[
\begin{align*}
X(i,j) &= (1.5 + 3(i/70.0)^5 + 4*\sin(j*\pi/200)^50)*\cos(4*i*\pi/175) \\
Y(i,j) &= (1.5 + 3(i/70.0)^5 + 4*\sin(j*\pi/200)^50)*\sin(4*i*\pi/175) \\
Z(i,j) &= j/50.0 + \cos(3*i*\pi/14)*\sin(j*\pi/1000)
\end{align*}
\]

v3d := draw(surface(X(i,j), Y(i,j), Z(i,j)), i=0..70, j=0..1000, _
style="smooth", title="Fisarmoniche")

colorDef(v3d, yellow(), yellow())

axes(v3d, "off")

zoom(v3d, 2.0, 2.0, 2.0)

Named after the accordion - whose bellows their bunched profiles recall - fisarmoniche are perfect for capturing thick sauces, which cling to their folds. This sturdy pasta is said to have been invented in the fifteenth century, in the Italian town of Loreto in the Marche, east central Italy.
5.32 Funghini

--- Funghini ---

\[
\begin{align*}
A(i,j) &= 5 \cos(i \pi/150) + 0.05 \cos(i \pi/3) \sin(j \pi/60)^{2000} \\
B(i,j) &= j/30.0 \ast (5 \sin(i \pi/150) + 0.05 \sin(i \pi/3)) \\
C(i,j) &= j/10.0 \ast (2 \sin(i \pi/150) + 0.05 \sin(i \pi/3)) \\
D(i,j) &= \begin{cases} 
B(i,j) & \text{if } (i \leq 150) \\
C(i,j) & \text{if } (j \leq 10) \\
2 \sin(i \pi/150) + 0.05 \sin(i \pi/6) & \text{else}
\end{cases} \\
X(i,j) &= 0.05 \cos(A(i,j) \pi/5) + 0.3 \cos(A(i,j) \pi/5) \sin(3D(i,j) \pi/50)^{2} \\
Y(i,j) &= 0.01 \sin(A(i,j) \pi/5) + 0.3 \sin(A(i,j) \pi/5) \sin(3D(i,j) \pi/50)^{2} \\
Z(i,j) &= 0.25 \sin((D(i,j) + 3) \pi/10) \\
v3d:=\text{draw(surface}(X(i,j),Y(i,j),Z(i,j)),i=0..300,j=0..30,\_ \\
style=\text{"smooth"},title=\text{"Funghini"}) \\
\text{colorDef}(v3d,\text{yellow()},\text{yellow()}) \\
\text{axes}(v3d,\text{"off"})
\end{align*}
\]

The modest dimensions of this \textit{pastine minute} (tiny pasta) make \textit{funghini} (little mushrooms) especially suitable for soups, such as a \textit{minestrone} made from chopped and sauteed celeriac.
A popular set from the pasta corta (short pasta) family, fusilli (little spindles) were originally made by quickly wrapping a spaghetto around a large needle. Best served as pastasciutta (pasta boiled and drained) with a creamy sauce containing slices of spicy sausage.
5.34 Fusilli al Ferretto

Fusilli al Ferretto

\[
\begin{align*}
A(i,j) &= 6.0\frac{i}{7.0+15\cos(j*\pi/20)} \\
X(i,j) &= (3+1.5\sin(i*\pi/140)^0.5\sin(j*\pi/20))*\sin(13*i*\pi/280) + _
\quad 5\sin(2*A(i,j)*\pi/135) \\
Y(i,j) &= (3+1.5\sin(i*\pi/140)^0.5\sin(j*\pi/20))*\cos(13*i*\pi/280) \\
Z(i,j) &= A(i,j)
\end{align*}
\]

\texttt{v3d:=draw(surface(X(i,j),Y(i,j),Z(i,j)),i=0..140,j=0..40,} \\
\texttt{style="smooth",title="Fusilli al Ferretto")} \\
\texttt{colorDef(v3d,yellow(),yellow())} \\
\texttt{axes(v3d,"off")} \\
\texttt{zoom(v3d,3.0,3.0,3.0,3.0)}

---

To create this Neapolitan variety of fusilli, a small amount of durum-wheat flour is kneaded and placed along a ferretto (small iron stick) that is then rolled between the hands to create a thick irregular twist of dough. The shape is removed and left to dry on a wicker tray known as a spasa. Fusilli al ferretto are best dished up with a lamb ragu.
A longer and more compact regional adaptation of fusilli, fusilli Capri are suited to a hearty ragu of lamb or por sausages, or may also be combined with rocket and lemon to form a lighter dish.
5.36 Fusilli Lunghi Bucati

\[ A(i,j) = 10 + \cos(i \times \pi/10) + 2 \cos((j+10) \times \pi/10) + 10 \cos((j+140) \times \pi/160) \]
\[ B(i,j) = 20 + \cos(i \times \pi/10) + 2 \cos((j+10) \times \pi/10) \]
\[ C(i,j) = (j+10.0) \times \pi/10.0 \]
\[ D(i,j) = i \times \pi/10.0 \]
\[ E(i,j) = 7 + 20 \sin((j-20) \times \pi/160) \]
\[ F(i,j) = 70 \times (0.1-(j-180.0)/200.0) \]
\[ X(i,j) = \begin{cases} A(i,j) & \text{if } (20 \leq j) \text{ and } (j \leq 180) \\ \cos(D(i,j)) + 2 \cos(C(i,j)) & \text{if } j \leq 20 \\ B(i,j) & \text{else} \end{cases} \]
\[ Y(i,j) = \begin{cases} \sin(D(i,j)) + 2 \sin(C(i,j)) & \text{if } (20 \leq j) \text{ and } (j \leq 180) \\ \sin(D(i,j)) + 2 \sin(C(i,j)) & \text{if } j \leq 20 \\ E(i,j) & \text{else} \end{cases} \]
\[ Z(i,j) = \begin{cases} E(i,j) & \text{if } (20 \leq j) \text{ and } (j \leq 180) \\ ((7.0 \times j)/20.0) & \text{if } j \leq 20 \\ F(i,j) & \text{else} \end{cases} \]
\[ v3d := \text{draw(surface}(X(i,j),Y(i,j),Z(i,j)),i=0..20,j=0..200,\text{style}="smooth",\text{title}="Fusilli Lunghi Bucati")} \]
\[ \text{colorDef}(v3d,\text{yellow()},\text{yellow()}) \]
\[ \text{axes}(v3d,\text{"off"}) \]
A distinctive member of the extended fusilli clan, fusilli lunghi bucati (long pierced fusilli) originated in Campania, southern Italy, and have a spring-like profile. Like all fusilli they are traditionally consumed with a meat-based ragu, but may also be combined with thick vegetable sauces and baked in an oven.
5.37 Galletti

According to their maker, Barilla, the origin of galletti (small cocks) is uncertain, but their shape recalls that of the chifferi with the addition of an undulating crest. Galletti are usually served in tomato sauces, but combine equally well with a boscaiola (woodsman’s) sauce of mushrooms.
5.38 Garganelli

A grooved pasta corta (short pasta), similar to maccheroni but with pointed slanting ends, garganelli are shaped like the gullet of a chicken (garganella in the northern Italian Emilian-Romagnolo dialect). Traditionally cooked in broth, garganelli are also sometimes served in hare sauce with chopped bacon.
5.39 Gemelli

--- Gemelli ---

\[
\begin{align*}
X(i,j) &= 6 \cos(j \cdot 1.9 \cdot \pi/50 + 0.55 \cdot \pi) \cdot \cos(3.0 \cdot i/25.0) \\
Y(i,j) &= 6 \cos(j \cdot 1.9 \cdot \pi/50 + 0.55 \cdot \pi) \cdot \sin(3.0 \cdot i/25.0) \\
Z(i,j) &= 8 \sin(j \cdot 1.9 \cdot \pi/50 + 0.55 \cdot \pi) + 3.0 \cdot i/4.0 \\
\end{align*}
\]

v3d := draw(surface(X(i,j), Y(i,j), Z(i,j)), i=0..100, j=0..50, _
style="smooth", title="Gemelli")
colorDef(v3d, yellow(), yellow())
axes(v3d, "off")
zoom(v3d, 2.5, 2.5, 2.5)

To create a gemello, a single pasta strand is twisted into a spiral with a deceptively dual appearance. In the south of Italy gemelli (twins) are served with tomato, mozzarella, and basil, while in the northwest they are preferred with pesto and green beans, or in salads.
With their fluted edges and cone-like shape, gigli resemble small bells (campanelle) or lilies (gigli), after which they are named. Another more recent design, gigli are shaped to capture thick meaty sauces.
5.41 Giglio Ondulato

Identical to gigli but with crenellated edges, *giglio ondulato* are made of durum-wheat flour and water.
5.42 Gnocchetti Sardi

As their name suggests, gnocchetti are simply small gnocchi (or 'dumplings'). They go well with ricotta or Pecorino Romano cheese, or served with thick sauces such as a veal ragù. Gnocchetti originated in Sardinia, a Mediterranean island to the west of Italy.
5.43 Gnocchi

Members of an extended family, gnocchi (dumplings) often resemble a semi-open grooved shell. Their preparation and ingredients (including potato, durum wheat, buckwheat and semolina) vary according to type. Gnocchi are often added to a sauce made from fontina cheese.
5.44 Gramigna

A speciality of the northern Italian region of Emilia-Romagna, gramigna (little weed) are traditionally served with a chunky sauce of sausages, or accompanied by the world-famous ragu alla bolognese. Alternatively, gramigna are sometimes presented alla pomodoro (with a light tomato sauce).
No longer than 15mm, lancette or 'hands' (of a clock) belong to the postine minute (tiny pasta) clan. They are delicious in consommes with a sprinkling of croutons and chopped greens. Lancette are also an excellent addition to mushroom or chicken soups.
As their name suggests, lasagna larga doppia riccia (large doubly curled lasagna) have two long undulating edges. The curls give the pasta a variable consistency when cooked, and help them to collect sauce. Lasagne are excellent with ricotta and a ragù napoletano, or cooked al forno (at the oven) with a creamy besciamella sauce.
5.47 Linguine

The thinnest member of the *bavette* (little dribble) family, *linguine* (little tongues) are best accompanied by fresh tomato, herbs, a drop of olive oil, garlic, anchovies, and hot peppers. *Linguine* may also be served with shellfish sauces, or white sauces of cream and soft cheese, flavoured with lemon, saffron, or ginger.
Originally from Campania and Liguria, *lumaconi rigati* (big ribbed snails) can be stuffed with a wide range of fillings, including spinach and ricotta cheese. Like cannelloni, they can then be covered in *besciamella* and cooked in the oven. Smaller members of the family (*lumache*) are also available.
The origin of the name maccheroni is uncertain. It is used generically to describe a hollow pasta corta (short pasta) that is made of durum-wheat flour and perhaps eggs. The pasta can be served con le sarde (with sardines) - a dish enhanced by a touch of fennel.
5.50 Maccheroni Alla Chitarra

Coming from the central Italian region of Abruzzo, maccheroni alla chitarra (guitar maccheroni) have a square cross-section, produced when a thin sheet of pasta is pressed through a frame of closely ranged wires (or chitarra) that lends the pasta its name. Usually served with a mutton ragu, or pallottelle (veal meatballs).
### 5.51 Mafaldine

Mafaldine are thin flat sheets of durum-wheat flour pasta with a rippled finish on each long edge. They are generally served with meaty sauces such as *ragù napoletano* or in seafood dishes.

---

Named at the turn of the twentieth century after Princess Mafalda of the House of Savoy, *mafaldine* are thin flat sheets of durum-wheat flour pasta with a rippled finish on each long edge. They are generally served with meaty sauces such as *ragù napoletano* or in seafood dishes.
One of the oldest-known durum-wheat varieties, manicotti (sleeves) were originally prepared by cutting dough into rectangles, which were topped with stuffing, rolled and finally baked al forno (at the oven). Today they are served containing a variety of cheeses and covered in
a savoury sauce, like filled dinner crepes.
This pasta is popular in the southeastern coastal region of Puglia in Italy, where it is customarily cooked in dishes with rapini, a relative of broccoli that grows plentifully in the area. Orecchiette (little ears) also pair well with other vegetables such as beans, and with salty seasonings such as anchovies, capers, or olives.
5.54 Paccheri

Part of the pasta corta (short pasta) family, paccheri are ribbed pasta cylinders that (due to their large size) are recommended with aubergine, seafood, or indeed any chunky sauce. It is thought the name stems from the term paccare, which means 'to smack' in the southern Italian region of Campania.
5.55 Pappardelle

This *pasta lunga* (long pasta) is often cooked with duck, pigeon, or other game fowl. *Pappardelle* are so popular that towns in Italy hold festivals in their honour, such as the *Sagra delle Pappardelle al Cinghiale* (Feast of the *Pappardelle* and Boar) in Torre Alfina, central Italy.
5.56 Penne Rigate

A versatile pasta, penne rigate (grooved quills) come from Campania, in southern Italy, and belong to the pasta corta (short pasta) family. They can be served with spicy arrabbiata (angry) sauce, which gets its name from the chillies and red peppers it contains.
5.57 Pennoni Lisci

--- Pennoni Lisci ---

\[
X(i,j) = \begin{cases} 
7 \sin(i \pi / 100)^2 & \text{if } (i < 100) \\
-7 \sin((i-100) \pi / 100)^2 & \text{else}
\end{cases}
\]

\[
Y(i,j) = \begin{cases} 
8 \cos(i \pi / 100) & \text{if } (i < 100) \\
-8 \cos((i-100) \pi / 100) & \text{else}
\end{cases}
\]

\[
Z(i,j) = \begin{cases} 
12 \cos(i \pi / 100) + 15 \sin((j-20) \pi / 40) & \text{if } (i < 100) \\
-12 \cos((i-100) \pi / 100) + 15 \sin((j-20) \pi / 40) & \text{else}
\end{cases}
\]

v3d := draw(surface(X(i,j), Y(i,j), Z(i,j)), i=0..200, j=0..40, 
style="smooth", title="Pennoni Lisci")

colorDef(v3d, yellow(), yellow())
axes(v3d, "off")
zoom(v3d, 2.0, 2.0, 2.0)

---

Similar in appearance to penne rigate, but larger and without the grooves, pennoni (large quills) require a more oily sauce (perhaps containing sliced chorizo) to cling to their smooth surface.
5.58 Pennoni Rigati

--- Pennoni Rigati ---

\[
\begin{align*}
A(i,j) &= -7\sin((i-100)\pi/100)^2 + 0.2\sin((3i-300)\pi/10) \\
B(i,j) &= -8\cos((i-100)\pi/100) + 0.2\cos((3i-300)\pi/10) \\
C(i,j) &= -12\cos((i-100)\pi/100) + 15\sin((j-20)\pi/40) \\
X(i,j) &= \begin{cases} 
7\sin(i\pi/100)^2 + 0.15\sin(3i\pi/10) & \text{if } i < 100 \\
A(i,j) & \text{else}
\end{cases} \\
Y(i,j) &= \begin{cases} 
8\cos(i\pi/100) + 0.15\cos(3i\pi/10) & \text{if } i < 100 \\
B(i,j) & \text{else}
\end{cases} \\
Z(i,j) &= \begin{cases} 
12\cos(i\pi/100) + 15\sin((j-20)\pi/40) & \text{if } i < 100 \\
C(i,j) & \text{else}
\end{cases}
\end{align*}
\]

\[v3d:=\text{draw(surface}(X(i,j),Y(i,j),Z(i,j)),i=0..200,j=0..40,_{
\text{style}=="smooth",\text{title}=="Pennoni Rigati"} {_
\text{colorDef}(v3d,\text{yellow}(),\text{yellow}()) \\
\text{axes}(v3d,"off") \\
\text{zoom}(v3d,2.0,2.0,2.0)\]]

The angled trim of the penne pasta family makes its members easy to recognize. Pennoni rigati (large grooved quills) are the ridged version of pennoni, and can be stuffed and cooked.
Another member of the *pastina minute* (tiny pasta) family, *puntalette* (tiny tips) are about 9 mm long, and no thicker than 3 mm. Like most *pastine*, they are best consumed in creamy soups, or perhaps in a salad.
5.60 Quadrefiore

An uncommon variety of pasta corta (short pasta), quadrefiori (square flowers) are sturdy, with rippled edges running down their lengths. Francis Ford Coppola, the maker and distributor, uses antique bronze moulds and wooden drying racks to achieve an authentic form and consistency.
These flat shapes are made with durum-wheat flour, eggs, and even nutmeg. Quadretti (tiny squares) are a classic pastine prepared using the leftovers of larger pasta sheets, and can be served as part of a traditional fish broth, or in soups containing fava beans. Their small shape means that they need only be cooked for a short time.
5.62 Racchette

--- Racchette ---

\[
A(i) = \sin(i\pi/2000)^{0.5}
\]

\[
X_0(i,j) = 2\cos((i+1500)\pi/1500) + 0.65\cos((i+750)\pi/1500) + \\
2\cdot(\text{abs}(\cos(i\pi/300)))^{100}\cos(i\pi/1500)
\]

\[
Y_0(i,j) = 2.4\sin((i+1500)\pi/1500) + 0.1\sin(i\pi/1500) + \\
2.3\cdot(\text{abs}(\sin(i\pi/300)))^{100}\sin(i\pi/1500)
\]

\[
X_1(i,j) = \\
\text{if } (i \leq 2000) \\
\text{then } 2.1\cos((2A(i)+1)\pi) + 0.65\cos((2A(i)+0.5)\pi) + \\
2.5\sin((A(i)+1.83)\pi)^{500} \\
\text{else } -2.1
\]

\[
Y_1(i,j) = \\
\text{if } (i \leq 2000) \\
\text{then } 2.5\sin((2A(i)+1)\pi) + 0.1\sin(A(i)\pi) + \\
3\sin((A(i)+1.83)\pi)^{500} \\
\text{else } 0.0
\]

\[
Z(i,j) = j/4.0
\]

vsp := createThreeSpace()
makeObject(surface(X0(i,j), Y0(i,j), Z(i,j)), i=0..3000, j=0..4, space==vsp)
makeObject(surface(X1(i,j), Y1(i,j), Z(i,j)), i=0..3000, j=0..4, space==vsp)
vp := makeViewport3D(vsp, style=="smooth", title=="Racchette")
colorDef(vp, yellow(), yellow())
axes(vp, "off")
Usually served in salads, *racchette* (rackets) suit crunchy pine nuts, sliced asparagus and fresh peas. Alternatively, the addition of diced watermelon or pomegranate seeds can create a light-tasking snack.
5.63 Radiatori

Small and squat, radiatori (radiators) are named after their ruffled edge. When boiled, drained, and served as a pastasciutta their open centre and large surface area holds thick sauces well, while the flaps sweep up and trap smaller morsels of food. This pasta is often accompanied by a lamb-, veal-, rabbit-, or pork-based ragù.
5.64 Ravioli Quadrati

--- Ravioli Quadrati ---

\[
X(i,j) = i/2.0 + 0.4 \sin((j+2.5)\pi/5) \times \\
(\sin(i\pi/200)^0.2 - \cos(i\pi/200)^0.2)
\]

\[
Y(i,j) = j/2.0 + 0.4 \sin((11i+25)\pi/50) \times \\
(\sin(j\pi/200)^0.2 - \cos(j\pi/200)^0.2)
\]

\[
Z(i,j) = \\
\text{if } \begin{cases} 
(10 < j \text{ and } j < 90) \text{ or } (10 < i \text{ and } i < 90) 
\text{ then } 10 \sin((i-10)\pi/80)^0.6 \sin((j-10)\pi/80)^0.6 
\text{ else if } \begin{cases} 
(10 > j) \text{ or } (10 > i) 
\text{ then } 0.0 
\text{ else } 0.0 
\end{cases}
\end{cases}
\]

v3d:=draw(surface(X(i,j),Y(i,j),Z(i,j)),i=0..99,j=0..99, \\
style=="smooth",title=="Ravioli Quadrati") 

colorDef(v3d,yellow(),yellow()) 

axes(v3d,"off") 

zoom(v3d,2.0,2.0,2.0)

---

Except for the square outline, *ravioli quadrati* (square ravioli) are made in an identical fashion to *ravioli tondi* (round ravioli). Other variations on the theme include crescents, triangles, and hearts. Some suggest the name *ravioli* derives from the verb meaning 'to wrap', others link it with *rapa*, the Italian word for turnip.
5.65 Ravioli Tondi

By far the best-known pasta ripiena (filled pasta), ravioli tondi (round ravioli) are made by sealing a filling between two layers of dough made from wheat flour and egg. Fillings vary enormously, from lavish pairings of meat and cheese, to more delicate centres of mushrooms, spinach, or even nettle.
5.66 Riccioli

A well-known pasta corta (short pasta), riccioli (curls) originated in the Emilia-Romagna region of northern Italy. Their ribbed exterior and hollow shape mean riccioli can retain a large quantity of sauce.
5.67 Riccioli al Cinque Sapori

Riccioli and riccioli al cinque sapori (curls in five flavors) are both members of the pasta corta (small pasta) family, without being related by structural similarities. Made of durum-wheat flour, riccioli al cinque sapori get their color from the addition of spinach, tomato, beetroot, and turmeric, and are usually served in broth.
Members of the pasta corta (short pasta) branch, and originally from southern Italy, rigatoni (large ridges) are very versatile. Their robust shape holds cream or tomato sauces well, but rigatoni are best eaten with sausages or game meat, mushrooms, and black pepper.
5.69 Rombi

A lesser-known pasta corta, rombi (rhombuses) feature two curled edges like lasagna doppia riccia, however, they are smaller and sheared on the diagonal. Generally served with sauce and pastasciutta or in brodo (in broth), according to size.
A modern design from the more unusual side of the pasta corta (short pasta) family, rotelle (small wheels) are constructed with spokes that help trap various flavors, making the pasta a good companion for a variety of sauces. Smaller versions can be served in salads or cooked in soups.
5.71 Saccottini

Another out-and-out member of the pasta ripiena (filled pasta) club, saccottini-like fagottini, to which they are closely related - are made of a durum-wheat circle of dough gathered into an irregular ball-shaped bundle. Saccottini are usually stuffed with ricotta, meat, or steamed greens.
5.72 Sagnarelli

This short and rectangular ribbon with four indented edges belongs to the pasta lunga (long pasta) branch. Sagnarelli are sometimes made with eggs and are generally served as pastasciutta with a meat ragu, or alongside vegetables such as wild asparagus.
5.73 Sagne Incannulate

Shaping the twisted ribbons of durum-wheat pasta known as sagne incannulate requires skill. Strips of dough are held at the end against a wooden board with one hand, while the palm of the other rolls the rest of the pasta lunga (long pasta) to form the distinctive curl. Sagne are best consumed with a thick sauce or a traditional ragu.
Hailing from the Amalfi coast in the province of Naples, scialatielli are a rustic pasta lunga (long pasta) similar in appearance to fettuccine and tagliotelle. When they are made, milk and eggs can be added to the durum-wheat flour to lend it a golden color, Scialatielli are best paired with seafood, or a **ragu** of pork or veal.
5.75 Spaccatelle

This noted speciality of Sicily belongs to the pasta corta (short pasta) family. Spaccatelle are generally elongated curves with a concave centre, and are served as a pastasciutta (pasta boiled and drained) with a light tomato sauce or a thick meaty ragù.
5.76  Spaghetti

Without a doubt, spaghetti (small strings) remain the best-known and most versatile pasta lunga (long pasta) worldwide. Above all they are known for accompanying ragu bolognese, a mixture containing beef, tomato, cream, onions, and pancetta. More recently, spaghetti have become popular in a creamy carbonara.
5.77 Spiralli

The ridged and helicoidal spirali (spirals) are similar in shape to cavatappi, but are slightly larger. Spirali may be served with chunky sauces as pastasciutta, baked al forno (at the oven) with a thick cheese topping or added to salads.
A member of the pastine minute (tiny pasta) family, stellette (little stars) are only marginally larger than both acini di pepe and cuoretti. Like all pastine, they are best served in a light soup, perhaps flavored with portobello mushrooms or peas.
5.79 Stortini

Another pastine minute (tiny pasta), stortini (little crooked pieces) are consumed in creamy soups with mushroom and celery. As a rule of thumb, smaller pasta is best in thinner soups,
while larger pastina minute can be served with thicker varieties.
The dough used to make strozzapreti (or strangolapreti; both translate to ‘priest strangle’) can be prepared with assorted flours, eggs, and even potato. Strozzapreti are an ideal pastasciutta, and can be served with a traditional meat sauce, topped with Parmigiano-Reggiano.
A mixture of wheat-flour and eggs, *tagliatelle* (derived from the Italian *tagliare* - 'to cut') belong to the *pasta lunga* (long pasta) family. Originally hailing from the north of Italy, *tagliatelle* are frequently served in *carbonara* sauce, but can also accompany seafood, or alternatively may form the basis of a *timballo* (baked pasta dish).
5.82 Taglierini

Another coiling pasta lunga (long pasta), taglierini are of the same lineage as tagliatelle, but are substantially narrower - almost hair-like.
5.83 Tagliolini

--- Tagliolini ---

\[
\begin{align*}
X_0(i,j) & = 0.5 \cos(i \pi/200) + 0.05 \cos(5i \pi/400) \\
Y_0(i,j) & = 0.5 \sin(i \pi/8000)^{0.1} \sin(i \pi/200) + 0.075 \sin(5i \pi/400) \\
Z_0(i,j) & = 0.01j + 0.1 \sin(i \pi/250) \\
X_1(i,j) & = 0.4 \cos(i \pi/200) \\
Y_1(i,j) & = 0.4 \sin(i \pi/8000)^{0.2} \sin(i \pi/200) \\
Z_1(i,j) & = 0.01j + 0.1 \sin(i \pi/250) \\
X_2(i,j) & = 0.3 \cos(i \pi/125) \\
Y_2(i,j) & = 0.3 \sin(3i \pi/2000) \sin(3i \pi/200) \\
Z_2(i,j) & = -0.05 + 0.01j + 0.1 \sin(i \pi/250)
\end{align*}
\]

\[
\text{vsp} := \text{createThreeSpace()}
\]

\[
\text{makeObject(surface}(X_0(i,j)*0.6,Y_0(i,j)*0.5,Z_0(i,j)),i=0..2000,j=0..1, _
\text{space} = \text{vsp})
\]

\[
\text{makeObject(surface}(X_1(i,j),Y_1(i,j),Z_1(i,j)),i=0..2000,j=0..2, _
\text{space} = \text{vsp})
\]

\[
\text{makeObject(surface}(0.8*X_1(i,j),0.9*Y_1(i,j),1.3*Z_1(i,j)),i=0..2000,j=0..2, _
\text{space} = \text{vsp})
\]

\[
\text{makeObject(surface}(X_2(i,j),0.9*Y_2(i,j),1.5*Z_2(i,j)),i=0..2000,j=0..2, _
\text{space} = \text{vsp})
\]

\[
\text{vp} := \text{makeViewport3D(vsp,style="smooth",title="Tagliolini")}
\]

\[
\text{colorDef(vp,yellow(),yellow())}
\]

\[
\text{axes(vp,"off")}
\]

Thinner even than taglierini, tagliolini are traditionally eaten as a starter with butter, soft cheese, or al pomodoro (in a light tomato sauce). Alternatively, they may be served in a chicken broth. A versatile pasta, tagliolini are also the main ingredient for a variety of
timballo dishes baked *al forno* (at the oven).
5.84 Torchietti

The whorls of *torchietti* (tiny torches) trap chunky sauces will. Also known as *maccheroni al torchio*, they are best eaten in a tomato sauce with larger, coarsely chopped vegetables.
such as carrots, broccoli, and cauliflower.
5.85  Tortellini

To prepare a tortellino, a teaspoon of meat, cheese, or vegetables is wrapped in a layer of dough (made of wheat flour and egg) that is then skillfully rolled and folded. These shapely members of the pasta ripiena (filled pasta) family are traditionally eaten in steaming soups, but are also drained and served with a local sauce as pastasciutta.
5.86 Tortiglioni

— Tortiglioni —

\[
\begin{align*}
X(i, j) &= 6 \cos(i \pi/75) - 3.5 \cos(j \pi/100) + \_ \\
&\quad + 0.15 \sin((13i/75.0+j/15.0+1.5) \pi) \\
Y(i, j) &= 6 \sin(i \pi/75) + 0.15 \sin((13i/75.0+j/15.0) \pi) \\
Z(i, j) &= 11.0 \cdot j/10.0
\end{align*}
\]

\vspace{0.5cm}

v3d:=draw(surface(X(i,j),Y(i,j),Z(i,j)),i=0..150,j=0..50, _
\quad style="smooth",title="Tortiglioni")

colorDef(v3d,yellow(),yellow())
axes(v3d,"off")
zoom(v3d,2.5,2.5,2.5)

Another classic *pasta corta* (short pasta), tortiglioni originate from the Campania region of southern Italy. Tortiglioni (deriving from the Italian torquere - 'to cut') are often baked in a *timballo* or boiled, drained and served as a *pastasciutta* coupled with a strong sauce of tomato, chorizo, and black pepper.
5.87 Trenne

\[ X(i,j) = \begin{cases} 
  3.0i/10.0 & \text{if } i \leq 33 \\
  (300.0 - 3i)/20.0 & \text{else}
\end{cases} \]

\[ Y(i,j) = \begin{cases} 
  0.0 & \text{if } i \leq 33 \\
  (9i - 300)/20.0 & \text{if } i \leq 66 \\
  (900 - 9i)/20.0 & \text{else}
\end{cases} \]

\[ Z(i,j) = \begin{cases} 
  \text{if } i \leq 16 \\
  (-9.0i/50.0 + 6.0j/4.0) & \text{then} \\
  (-6.0 + 9.0i/50.0 + 3.0j/2.0) & \text{else}
\end{cases} \]

\[ \text{colorDef(v3d,yellow(),yellow())} \]
\[ \text{axes(v3d,"off")} \]
\[ \text{zoom(v3d,4.0,4.0,4.0)} \]
A hollow triangular variety of pasta corta (short pasta), trenne (penne with a triangular cross-section) are extremely sturdy. They are best served with mushrooms, tomato, and spinach - or in any sauce that would normally accompany their close cousins: penne, trennette, and ziti.
5.88 Tripoline

The ribbon-like *tripoline* are *pasta lunga* (long pasta) curled along one edge only. This wave gives the *tripoline* a varying texture after cooking, and helps them to gather extra sauce. Originally from southern Italy, they are often served with tomato and basil, or *ragu alla napoletana* and a sprinkling of Pecorino Romano.
5.89 Trofie

A(i,j) == (3.0*i)/5.0+10*cos(j*%pi/25)
X(i,j) == (1+sin(i*%pi/150)+2*sin(i*%pi/150)*sin(j*%pi/25)) * _
   sin(13*i*%pi/300)
Y(i,j) == (1+sin(i*%pi/150)+2*sin(i*%pi/150)*sin(j*%pi/25)) * _
   cos(13*i*%pi/300)+5*sin(2*A(i,j)*%pi/125)
Z(i,j) == A(i,j)
v3d:=draw(surface(X(i,j),Y(i,j),Z(i,j)),i=0..150,j=0..50,_
   style=="smooth",title=="Trofie")
colorDef(v3d,yellow(),yellow())
axes(v3d,"off")
zoom(v3d,3.0,3.0,3.0)

The Ligurian version of gnocchi, trofie are made from a mixture of wheat flour, bran, water, and potatoes. Traditionally served boiled with green beans and more potatoes, they may also be paired with a simple mix of pesto genovese, pine nuts, salt, and olive oil.
5.90 Trottole

A well-formed *pasta corta* (short pasta) comprised of rings that curl up about a central stalk, *trottole* are ideal for salads. They are also delicious with pumpkin or courgette, leek, pine...
nuts, and a few shavings of Parmigiano-Reggiano.
The smallest members of the *pasta corta* (short pasta) clan, *tubetti rigati* (grooved tubes) were first created in Campania, southern Italy. When served *e fagioli* (with beans) they create very filling soups, but can also be served in a light *marinara* (mariner's) sauce of tomato, basil, and onions.
A pasta reserved for banquets and special occasions, ziti ('grooms' or 'brides' in Italian dialect) originate from Sicily. Tradition has it that they should be broken by hand before being tossed into boiling water. After draining they can be served in tomato sauces with peppers or courgettes, topped with cheese like Provolone.
Bibliography


   http://axiom.axiom-developer.org


[6] Daly, Timothy, "The Axiom Literate Documentation"
   http://axiom.axiom-developer.org/axiom-website/documentation.html

Chapter 6

Index
Index

figures
CRCp26-2.1.1.1-3, 51
CRCp26-2.1.1.4-6, 52
CRCp26-2.1.2.1-3, 54
CRCp26-2.1.2.4-6, 55
CRCp28-2.1.3.1-6, 56
CRCp28-2.1.3.7-10, 58
CRCp28-2.1.4.1-6, 59
CRCp28-2.1.4.7-10, 61
CRCp30-2.2.1.1-3, 62
CRCp30-2.2.2.1-3, 63
CRCp30-2.2.3.1-3, 65
CRCp30-2.2.4.1-3, 66
CRCp30-2.2.5.1-3, 67
CRCp30-2.2.6.1-3, 68
CRCp32-2.2.10.1-3, 73
CRCp32-2.2.11.1-3, 75
CRCp32-2.2.12.1-3, 76
CRCp32-2.2.7.1-3, 70
CRCp32-2.2.8.1-3, 71
CRCp32-2.2.9.1-3, 72
CRCp34-2.2.13.1-3, 77
CRCp34-2.2.14.1-3, 78
CRCp34-2.2.15.1-3, 80
CRCp34-2.2.16.1-3, 81
CRCp34-2.2.17.1-3, 82
CRCp34-2.2.18.1-3, 84
CRCp36-2.2.19.1-3, 85
CRCp36-2.2.20.1-3, 86
CRCp36-2.2.21.1-3, 88
CRCp36-2.2.22.1-3, 89
CRCp36-2.2.23.1-3, 90
CRCp36-2.2.24.1-3, 92
CRCp38-2.2.25.1-3, 93
CRCp38-2.2.26.1-3, 94
CRCp38-2.2.27.1-3, 96
CRCp38-2.2.28.1-3, 97
CRCp38-2.2.29.1-3, 98
CRCp38-2.2.30.1-3, 100
CRCp40-2.2.31.1-3, 101
CRCp40-2.2.32.1-3, 102
CRCp40-2.2.33.1-3, 104
CRCp42-2.3.1.1-3, 105
CRCp42-2.3.2.1-3, 106
CRCp42-2.3.3.1-3, 108
CRCp42-2.3.4.1-3, 109
CRCp44-2.3.5.1-3, 110
CRCp44-2.3.6.1-3, 112
CRCp44-2.3.7.1-3, 113
CRCp44-2.3.8.1-3, 114
CRCp46-2.4.1.1-3, 116
CRCp46-2.4.2.1-3, 117
CRCp46-2.4.3.1-3, 118
CRCp46-2.4.4.1-3, 120
CRCp48-2.4.5.1-3, 121
CRCp48-2.4.6.1-3, 122
CRCp48-2.4.7.1-3, 123
CRCp48-2.4.8.1-3, 125
CRCp50-2.5.1.1-3, 126
CRCp50-2.5.2.1-3, 127
CRCp50-2.5.3.1-3, 129
CRCp50-2.5.4.1-3, 130
CRCp50-2.5.5.1-3, 131
CRCp50-2.5.6.1-3, 132
CRCp52-2.6.1.1-3, 134
CRCp52-2.6.2.1-3, 135
CRCp52-2.6.3.1-3, 136
CRCp52-2.6.4.1-3, 138
CRCp52-2.6.5.1-3, 139
CRCp52-2.6.6.1-3, 140
CRCp54-2.7.1.1-3, 142
CRCp54-2.7.2.1-3, 143
CRCp54-2.7.3.1-3, 144
CRCp54-2.7.4.1-3, 146
CRCp54-2.7.5.1-3, 147
CRCp54-2.7.6.1-3, 148
CRCp56-2.8.1.1-3, 150
CRCp56-2.8.2.1-3, 151
CRCp56-2.8.3.1-3, 152
CRCp56-2.8.4.1-3, 154
CRCp56-2.8.5.1-3, 155
CRCp56-2.8.6.1-3, 156
CRCp58-2.9.1.1-3, 158
CRCp58-2.9.2.1-3, 159
CRCp58-2.9.3.1-3, 160
CRCp58-2.9.4.1-3, 161
CRCp58-2.9.5.1-3, 162
CRCp58-2.9.6.1-3, 164
CRCp60-2.9.10.1-3, 168
CRCp60-2.9.11.1-6, 170
CRCp60-2.9.7.1-3, 165
CRCp60-2.9.8.1-3, 166
CRCp60-2.9.9.1-3, 167