The *Mathematica*® Journal

Volume 8, Issue 4

Contents

Departments

Staff and Contributors 490
Calendar 491
Announcement 495
News Bulletins 498
New Products 500
New in the Library 509
New Publications 510
Article Submissions to *The Mathematica Journal* 599
Classifieds 603

Tutorials

Tricks of the Trade *edited by Paul Abbott* 516
In and Out *edited by Paul Abbott* 523

Columns

Trott's Corner *by Michael Trott* 532

Feature Articles

Demystifying Rules *by Nancy Blachman* 563
Animated Three-Dimensional Structure of the Earth's Interior for Research and Education *by Hiroki Sato, Kenichi Muro, and Akira Hasegawa* 574
How to Color a Graph with (Computer) Algebra *by Yuri Matiyasevich* 586
Clipping Polygons *by Garry Helzer* 591

The electronic version of this issue is available at www.mathematica-journal.com/issue/v8i4.

The online edition of *The Mathematica Journal* includes downloadable versions of all articles and packages in this issue.

On the Cover

The cover shows the Riemann surface $w(z) = \frac{\log(z^2 - i)}{\sqrt[3]{z^3 - 1}}$. For more examples, see Michael Trott's article *Visualization of Riemann Surfaces IId.*

Staff and Contributors

Managing Editor: George Beck
Project Manager: Louise Holubek
Design Team: Jeremy Davis, Jody Jasinski
Compositor: Glenn Scholebo
Archives and Web Production: Jean Buck
Web Development: Andy Hunt, Lori Goodman
Editorial Staff: George Beck, Marcia Krause, Marian Peden
Customer Service: Matt Doman, Alissa Urie

Columnists: Paul Abbott, Michael Trott
Contributors: Nancy Blachman, Akira Hasegawa, Garry Helzer, Yuri Matiyasevich, Kenichi Muro, Hiroki Sato

Editorial Advisory Board: Steve Christensen (MathTensor, Inc.; Christensen and Associates, Inc.); Donald Glaser (University of California, Berkeley); Donald Knuth (Stanford University); Wolfram Koepf (Konrad-Zuse-Zentrum, Berlin); Silvio Levy (Mathematical Sciences Research Institute); David Lovelock (University of Arizona); Jerry Marsden (Caltech); Robert Moore (University of Washington, Seattle); Alfred van der Poorten (Macquarie University); Bill Press (Harvard University); Sam Savage (University of Chicago); Dana Scott (Carnegie Mellon University); Lawrence M. Seiford (University of Michigan); Lynn Steen (St. Olaf College); Francis Sullivan (Institute for Defense Analyses); Bill Thurston (University of California, Davis); Hal Varian (University of California, Berkeley); Rob Wolff (Apple Computer, Inc.); and Stephen Wolfram (Wolfram Research, Inc.).

Customer Services
www.mathematica-journal.com
+1-877-294-8184 (U.S. and Canada)
+1-217-398-6000
fax: +1-217-398-1108
subscriptions: subscribe@mathematica-journal.com

Advertising
The Mathematica Journal accepts advertising for all *Mathematica*-related products and services. Classified advertising is provided for *Mathematica* users. Rates are available at www.mathematica-journal.com/advertising. Contact the advertising department through the telephone numbers above, or send email to ads@mathematica-journal.com.

The Mathematica Journal is typeset in *Mathematica* 4 using Janson, Futura, Math CMTT and the standard *Mathematica* fonts.

Publishing
The Mathematica Journal (ISSN 1047-5974) is published quarterly by Wolfram Media, Inc. Annual subscription rates in U.S. dollars: Electronic version: individuals, $35; institutions, $70; students, $20. Electronic and print versions: individuals, $75; institutions, $200; students, $50. Canadian/Mexican print subscriptions, add $10. Print subscriptions outside North America, add $32. Single print copy and back issue sales price: $25. To subscribe or renew, call +1-217-398-6000 or +1-877-294-8184 (U.S. only), or visit our website at www.mathematica-journal.com to enter your subscription information. Postmaster: Send address changes to: *The Mathematica Journal*, P.O. Box 6359, Champaign, IL 61826-6359.

© 2002 Wolfram Media, Inc. All rights reserved. No part of this publication may be reprinted or otherwise reproduced without the express permission of its publisher. No copyright claimed for U.S. government materials.

Mathematica, *MathLink*, and *MathSource* are registered trademarks of Wolfram Research, Inc. *Mathematica* is not associated with Mathematica Policy Research, Inc. or MathTech, Inc. All other product names mentioned are trademarks of their producers. Material in this publication may not be reproduced in any form without permission of the publisher. The opinions expressed in the Journal are the authors' own and do not necessarily reflect those of the authors' employers, the editors, or the publishers of the *Journal*.

Calendar

Professional meetings, conferences, trade shows, exhibits, and *Mathematica* training courses.

Send your event notices to calendar@mathematica-journal.com.

August

August 20, 2002
Introduction to *Mathematica*
CANdiensten
Amsterdam, Netherlands
Info: www.candiensten.nl/english/cursussen/cursussendetail.asp?id=8

August 29–30, 2002
Numerical Methods for Pricing Financial Derivatives
CANdiensten
Amsterdam, Netherlands
Info: www.candiensten.nl/english/cursussen/cursussendetail.asp?id=25

September

September, 2002
M101: A First Course in *Mathematica*
Wolfram Education Group
September 16–17, Los Angeles, California
September 16–17, Washington, DC
September 23–24, Austin, Texas
September 23–24, San Francisco, California
Info: www.wolfram.com/services/education/courses/calendar.html

M110: Working with Data Using *Mathematica*
Wolfram Education Group
September 18, Seattle, Washington
Info: www.wolfram.com/services/education/courses/calendar.html

September 19, 2002
web*Mathematica* Seminar
Wolfram Research
Washington, DC
Info: www.wolfram.com/services/seminars/dc2002

September 26, 2002
Introduction to *Mathematica*
CANdiensten
Amsterdam, Netherlands
Info: www.candiensten.nl/english/cursussen/cursussendetail.asp?id=8

October

October, 2002

M101: A First Course in *Mathematica*

Wolfram Education Group

October 7–8, Boston, Massachusetts
October 7–8, Detroit, Michigan
October 21–22, Chicago, Illinois
October 21–22, New York, New York
October 21–22, San Jose, California

Info: www.wolfram.com/services/education/courses/calendar.html

M220: Programming in *Mathematica*

Wolfram Education Group

October 7–8, Chicago, Illinois
October 10–11, Boston, Massachusetts
October 14–15, New York, New York
October 17–18, Washington, DC

Info: www.wolfram.com/services/education/courses/calendar.html

October 10–11, 2002

Financial Modeling with *Mathematica*

CANdiensten

Amsterdam, Netherlands

Info: www.candiensten.nl/english/cursussen/cursussendetail.asp?id=17

October 29–30, 2002

***Mathematica* Intermediate and Programming Course**

CANdiensten

Amsterdam, Netherlands

Info: www.candiensten.nl/english/cursussen/cursussendetail.asp?id=9

November

November, 2002

M101: A First Course in *Mathematica*

Wolfram Education Group

November 7–8, Minneapolis, Minnesota
November 7–8, Raleigh, North Carolina
November 7–8, Washington, DC
November 13–14, San Francisco, California

Info: www.wolfram.com/services/education/courses/calendar.html

M220: Programming in *Mathematica*

Wolfram Education Group

November 7–8, San Francisco, California

Info: www.wolfram.com/services/education/courses/calendar.html

M110: Working with Data Using *Mathematica*
Wolfram Education Group
November 15, Chicago, Illinois
November 15, San Francisco, California
Info: www.wolfram.com/services/education/courses/calendar.html

November 14, 2002
Introduction to *Mathematica*
CANdiensten
Amsterdam, Netherlands
Info: www.candiensten.nl/english/cursussen/cursussendetail.asp?id=8

November 21, 2002
***Mathematica* 4.2 User's Conference**
CANdiensten
Amsterdam, Netherlands
Info: www.candiensten.nl/english/nieuwsevenement/nieuwsevenementdetail.asp?id=65

November 22, 2002
***Mathematica* 4.2 User's Conference**
CANdiensten
Brussels, Belgium
Info: www.candiensten.nl/english/nieuwsevenement/nieuwsevenementdetail.asp?id=65

December

December 10–11, 2002
Mastering *Mathematica*
Ritme Informatique
Paris, France
Info: www.ritme.com/formation/mathematica

December 12, 2002
Programming with *Mathematica*
Ritme Informatique
Paris, France
Info: www.ritme.com/formation/mathematica

December 19–20, 2002
Mathematica Intermediate and Programming Course
CANdiensten
Amsterdam, Netherlands
Info: www.candiensten.nl/english/cursussen/cursussendetail.asp?id=9

ANNOUNCEMENT

Stephen Wolfram's A New Kind of Science

May 14, 2002, saw the publication of Stephen Wolfram's book *A New Kind of Science*—the culmination of more than twenty years of work by scientist, *Mathematica* creator, and Wolfram Research CEO Stephen Wolfram. An instant bestseller, the book is being widely read by scientists and non-scientists alike.

Mathematica played several crucial roles in the creation of *A New Kind of Science* (abbreviated NKS). Most obviously, *Mathematica* is the notation used in the technical notes in the book—which contain nearly 800 *Mathematica* programs. *Mathematica* was also extensively used in the physical production of the book—notably in the creation of the nearly 1000 striking computer graphics that fill the book.

But behind the scenes *Mathematica* was even more central: it was the main tool that made it possible for the science in NKS to be developed at all.

Stephen Wolfram began the research that eventually led to NKS in the early 1980s. But by the mid-1980s his work was blocked by the lack of adequate computational tools. This was a large part of the reason that he decided to develop *Mathematica*.

As Wolfram describes it, having *Mathematica* allowed him to do in only ten years what might otherwise have taken fifty. Wolfram likens one of the roles of *Mathematica* in NKS to early uses of the telescope. For just as the telescope suddenly opened up a whole host of discoveries in the astronomical world, so *Mathematica* suddenly opened up a whole host of new discoveries in the computational world of simple programs.

In NKS, Wolfram discusses the connections and implications of his discoveries for a very wide range of issues in mathematics, physics, biology, computer science, and elsewhere. And throughout, he uses *Mathematica* as a crucial tool to apply both existing and new methods of analysis.

After a break of several years for the development of Versions 1 and 2 of *Mathematica*, Wolfram began to work in earnest on NKS in 1991. And over the next ten years he used *Mathematica* intensively every day in his research, altogether running more than a million lines of *Mathematica* input.

 © 2002 Wolfram Media, Inc.

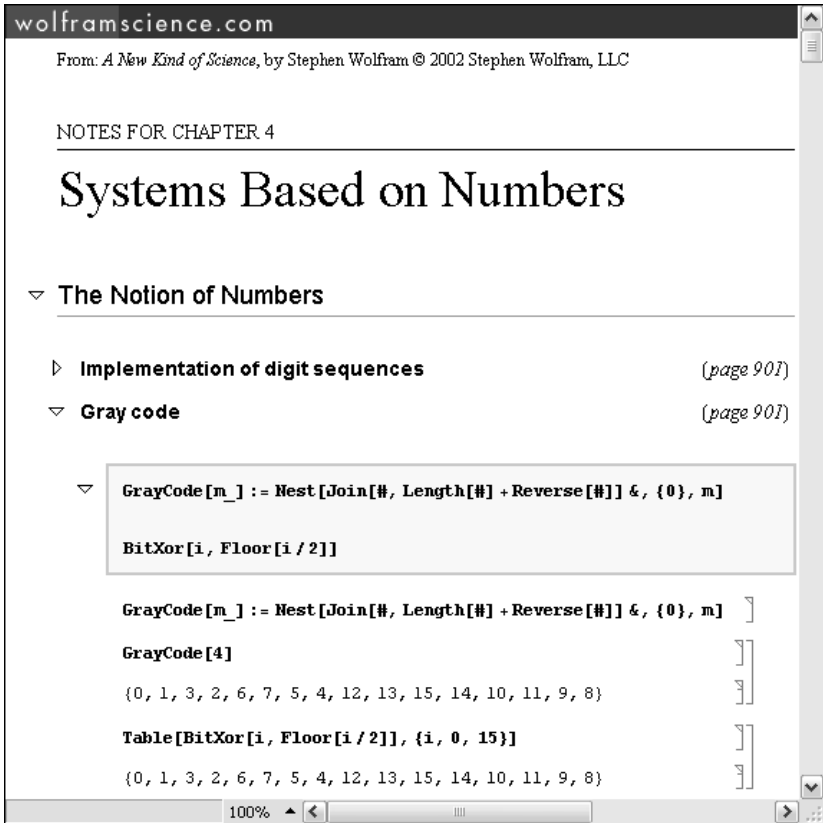
The Programs of NKS

In the main text of NKS Wolfram explains his ideas and discoveries using just ordinary language and pictures. But in the extensive technical notes at the back of the book he uses more technical formalism. And throughout, everything is represented by *Mathematica* programs.

NKS is in many ways a remarkable showcase of the power of the *Mathematica* language. The 800 or so programs in NKS cover a huge range of computational functions and concepts—with extremely short and clear programs that make use of the many programming paradigms available in *Mathematica*.

Mathematica users at all levels will find much to study in the construction and operation of the programs in NKS. Subsequent issues of this journal will contain articles that dissect and discuss some of the programs in NKS. The index to NKS (available online at www.wolframscience.com/nks/index) lists many *Mathematica* functions, showing where they are used in the programs.

All the programs in the notes to NKS are available online at www.wolframscience.com/nks/programs. The programs are presented in notebooks (see example below). In each case, the original form of the program as it appears in the printed version of NKS is given, together with test examples and a version conveniently set up for experimentation.



A NEW KIND OF SCIENCE™ Explorer

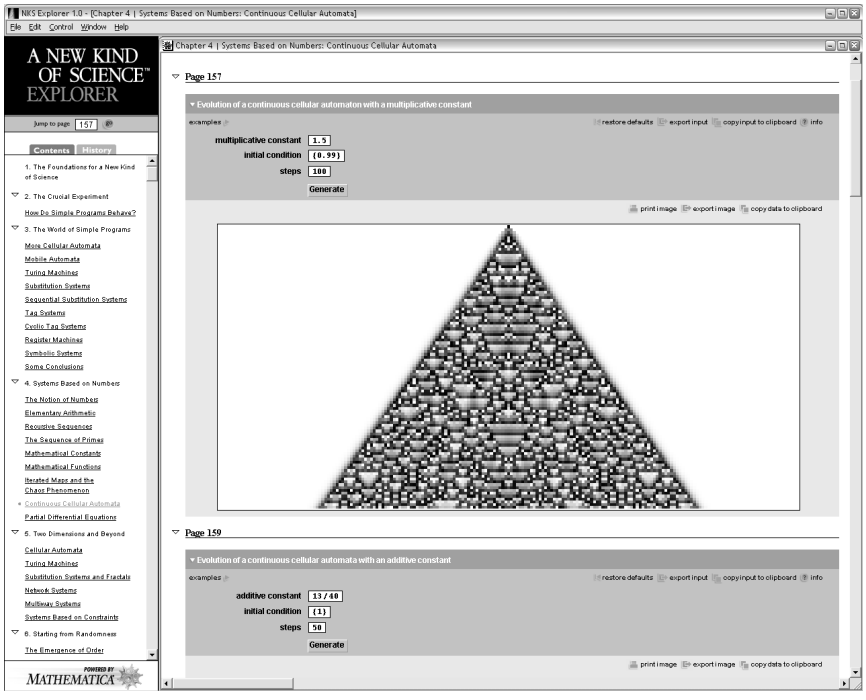
Coming soon is a new *Mathematica*-powered product (illustrated below) that allows anyone to reproduce the key graphics from NKS on their own computers, and to do their own experiments on the discoveries in NKS.

NKS Explorer is based on the very same *Mathematica* programs that Stephen Wolfram used to create the graphics in his book. But using the latest custom *Mathematica* technology, Wolfram Research will be delivering *NKS Explorer* as a standalone product.

The form-based interface to *NKS Explorer* is exclusively built using notebooks and *Mathematica*'s symbolic document capabilities. The simplicity of the interface makes *NKS Explorer* ideal for a very wide range of users—from experienced scientists to elementary school children.

NKS Explorer—like other *Mathematica*-powered products such as *CalculationCenter*—serves as an interesting model for interface development in *Mathematica*.

NKS Explorer is set up to be able to exchange data with *Mathematica*. Also coming soon is *NKS Explorer Mathematica Kit*, which allows functions from *NKS Explorer* to be run within a *Mathematica* session.



For more information about NKS, or to order the book, visit www.wolframscience.com.

For more information about NKS Explorer, or to purchase the software, visit www.wolframscience.com/nksx.

News Bulletins

View more *Mathematica* news at www.wolfram.com/news.

Mathematica Author and Publisher Program

The *Mathematica* Author and Publisher Program is designed to assist authors and publishers of *Mathematica*-related books with support, tools, and information for book authoring, publishing, and marketing. More details are available at www.wolfram.com/company/programs/author.

Mathematica Speaker Assistance Program

The *Mathematica* Speaker Assistance Program makes a number of Wolfram Research resources more easily available to trainers and consultants for their *Mathematica*-related presentations and seminars. Program benefits include informative promotional materials, calendar listings, and possibly other publicity on the Wolfram Research website. For additional information, visit www.wolfram.com/services/training/speakers.

MathML Central: A Wolfram Web Resource

MathML Central, a new Wolfram web resource that uses web*Mathematica* technology, is the place to go for information about MathML. The MathML Central website contains extensive examples, documentation, tutorials, and other tools. The MathML tools on the site take advantage of *Mathematica*'s built-in capabilities for importing, processing, and exporting MathML. Explore this new resource at www.mathmlcentral.com.

MathML International Conference 2002

Details and transcriptions of select conference talks and presentations at the MathML International Conference 2002, hosted by Wolfram Research from June 28 to 30 in Chicago, Illinois, are available online at www.mathmlconference.org/2002.

Finance Data Feed Table

The table showing various ways to link Bloomberg and Reuters data feeds with *Mathematica* can be found at www.wolfram.com/solutions/finance/datafeed.html.

HostSRV.com

HostSRV.com is a pioneer in the business of hosting sites that use web*Mathematica* to instantly perform complex calculations. Says founder Tom Compton, "For a small subscription fee, users avoid systems administration issues and have tech support to get over troublesome problems. Additionally, their sites benefit from use of a dedicated server and access to a pool of kernels." For more information, see www.wolfram.com/news/hostsrv.html.

MathXL for Calculus

Addison Wesley, one of the world's most respected textbook publishers, has launched new online calculus tutorials and testing using web*Mathematica* technology from Wolfram Research. MathXL for Calculus is a web-based program designed to help diagnose students' calculus skills and to create a personalized study plan for each student based on the student's test results. Learn more about MathXL for Calculus at www.wolfram.com/news/mathxl.html.

Smithsonian National Museum of American History

A new Smithsonian National Museum of American History exhibit on teaching math in America features *Mathematica* as a key teaching tool. The information on *Mathematica* on the museum website can be located by visiting americanhistory.si.edu/teachingmath, clicking "The Information Age" in the sidebar, and proceeding to page two.

web*UnRisk*

web*UnRisk*, an interactive new risk analysis tool on the Wolfram Research website, is just one example of the types of financial analysis you can perform when you combine the online technical computing power of web*Mathematica* with a top-notch derivatives pricing engine like *UnRisk*. You can use the sample data given or upload your own. Read more about this interactive tool at www.wolfram.com/news/webunrisk.html.

New Products

Mathematica 4.2

New *Mathematica* 4.2 delivers XML and Java functionality.

With the release of *Mathematica* 4.2, Wolfram Research, Inc. introduced the first technical computing system with full XML support and seamless Java integration. Now users can access any Java program from within *Mathematica* and work directly with XML data and documents. Version 4.2 also brings important advancements in linear programming, statistics, optimization, combinatorics, graph theory, and technical publishing.

Key new features in *Mathematica* 4.2 include the following:

- Transparent Java integration with *J/Link* 2.0 and built-in Java Runtime Engine
- XML extensions that allow *Mathematica* notebooks and expressions to be stored as XML
- New bundled XML tools package for symbolic XML manipulation
- Support for XHTML export including style sheets
- Extended MathML 2.0 support
- Improved linear programming and optimization
- Statistics enhancements including new ANOVA package
- New bundled *Combinatorica* package for combinatorics and graph theory
- New bundled *AuthorTools* package for technical publishing
- Slide show environment for presentations
- New import and export formats including FITS and SDTS

"The built-in Java Runtime Environment and *J/Link* 2.0 give developers an easy way to mix *Mathematica* and Java code right out of the box," says Lars Hohmuth, Product Manager for *Mathematica*. "Whether they want to build graphical user interfaces to *Mathematica* using AWT and Swing graphical user interface components or connect to their enterprise database using JDBC, they can now use Java with the same ease as the built-in *Mathematica* programming language."

Version 4.2 also marks the beginning of the XML era at Wolfram Research by offering support for import, export, and symbolic manipulation of XML documents and data. In conjunction with a major focus on technical publishing in this version, *Mathematica* now includes XML extensions that let users save notebooks

and *Mathematica* expressions as XML constructs and adds support for XHTML and MathML 2.0, the emerging standard for mathematics on the web.

Mathematica 4.2 also includes a new *AuthorTools* package and a slide show environment that make preparing notebooks for presentations and online and print publication much easier.

"Version 4.2 takes the changing *Mathematica* customer base into account without neglecting our core users," says Roger Germundsson, Director of Research and Development at Wolfram Research. "An increasing number of our customers are using *Mathematica* in nontraditional ways—for example, as a metaprogramming tool, for document processing, or as a deployment mechanism—but we are also still committed to our core competency in technical computing. *Mathematica* 4.2 contains the largest collection of mathematical algorithms worldwide and is the only major technical computing system that includes state-of-the-art technology for global optimization, combinatorics, and graph theory in the core product."

"*Mathematica* has always been uniquely capable as the hub of a technical professional's world," adds Conrad Wolfram, Director of Strategic Development. "Version 4.2 makes it ideal as an enterprise's technical hub too. The integration of Java, XML, and *Mathematica*'s capabilities in 4.2 holds the key to computational and document-oriented processing across an organization."

Mathematica 4.2 is available for Windows, Mac OS, Mac OS X, Linux (PC, Alpha, PowerPC), Solaris, HP-UX, IRIX, AIX, Compaq Tru64 Unix, and compatible systems.

Mathematica 4.2 began shipping in the United States and Canada June 24. It was available in all other territories except Japan starting July 8. The Japanese edition of *Mathematica* 4.2 is scheduled for release in early fall 2002. More product details are available on the Wolfram Research website.

More info
www.wolfram.com/products/mathematica/newin42

web*Mathematica* for Mac OS X

Wolfram Research introduces web*Mathematica* for Mac OS X.

Wolfram Research, Inc. announces that web*Mathematica*, the revolutionary product that merges the computational power of *Mathematica* with the convenience and accessibility of the web, is now available for Mac OS X.

web*Mathematica* provides the ability to quickly and easily add interactive calculations to the web, enabling users to:

- Build custom websites that provide specialized calculations
- Compute, visualize, and analyze data using a web browser
- Deliver sophisticated courseware and publish interactive books

- Provide active functionality for technical documentation that can be instantly updated from a centralized location
- Integrate computation into existing enterprise infrastructure

"While web*Mathematica* has already made a major impact on the paradigm for delivering technical computations, we expect another major boost in applications now that web*Mathematica* is available for Mac OS X," says Lars Hohmuth, Product Manager for web*Mathematica* at Wolfram Research. "The Macintosh has traditionally been the platform for innovative and creative people, so we are looking forward to seeing what they come up with."

"Mac OS X provides a powerful and feature-rich platform for performing intensive, scientific computing," says Ron Okamoto, Apple's vice president of Worldwide Developer Relations. "We're excited to have web*Mathematica* available for Mac OS X because it gives our users the remarkable ability to create web applications that can conduct interactive calculations and visualization directly within a web browser."

Since its initial release, web*Mathematica* has generated extensive interest with a number of banks, engineering firms, and other institutions already starting to phase it into their enterprisewide systems. It has also caught on quickly with individual users, hundreds of whom are now using it to express their creativity or to work in new ways over the internet.

"The solid Unix foundations of Mac OS X and the included Java 2 Standard Edition, a unique feature among operating systems, make Mac OS X an ideal platform for web*Mathematica*," says Theodore Gray, cofounder of Wolfram Research and chief architect of the *Mathematica* user interface. "In addition, the ease-of-use features in Mac OS X Server make setting up and administering a high-performance web server a straightforward proposition. The Mac platform is also renowned for the availability of best-of-class software solutions like Photoshop and QuickTime Streaming Server for the generation of immersive websites."

web*Mathematica* is built on Java servlets, making it compatible with any web server, servlet engine, or application server that supports the Servlet 2.0 or later API and giving developers a broad choice of web solutions to work from.

web*Mathematica* is also available for other major operating systems and platforms including Windows, Linux, and Solaris. It comes in two license configurations: web*Mathematica Professional*, intended for the development and deployment of sophisticated intranet and internet sites, and web*Mathematica Amateur*, designed for individuals who wish to showcase their work over the web. Subscribers to Wolfram Research's *Premier Service* can receive complimentary copies of web*Mathematica Amateur*.

Information on ordering web*Mathematica* is available online. Additional information about web*Mathematica*, including working web*Mathematica* examples, is also available.

More info
www.wolfram.com/products/webmathematica

Japanese web*Mathematica*

web*Mathematica* Japanese Edition is a unique product that brings computation to the web.

Wolfram Research, Inc. has released web*Mathematica Japanese Edition*, a revolutionary product that merges the computational power of *Mathematica* with the convenience and accessibility of the web.

web*Mathematica* provides the long sought-after ability to quickly and easily add interactive calculations to the web, enabling users to:

- Build custom websites that provide specialized calculations
- Compute, visualize, and analyze data using a web browser
- Deliver sophisticated courseware and publish interactive books
- Provide active functionality for technical documentation that can be instantly updated from a centralized location
- Integrate computation into existing enterprise infrastructure

"web*Mathematica* changes the paradigm for delivering technical computations," says Lars Hohmuth, Product Manager for web*Mathematica* at Wolfram Research. "Using web*Mathematica*, companies and research and educational institutions can rapidly create customized applications and deploy them over the web, using just standard web tools and a working knowledge of *Mathematica*." web*Mathematica* generated unprecedented international interest during its beta test phase, and many banks, engineering firms, and other institutions are already starting to phase it into their enterprisewide systems.

"web*Mathematica* satisfies our long-standing ambition to bring truly interactive calculations to the web," says Conrad Wolfram, Director of Strategic and International Business Development. "Judging by the dramatic way in which testers are adopting the technology, I'd say it fulfills their needs too." Says Cheng Wu of eFunda, "We are extremely impressed with web*Mathematica*'s performance, especially its rock-solid stability. In the race of bringing world-class mathematics software to the web, Wolfram Research is way ahead of its competition."

web*Mathematica* is built on Java servlets, making it compatible with any web server, servlet engine, or application server that supports the Servlet 2.0 or later API and giving developers a broad choice of platforms and web solutions to work from.

"Our adoption of modern web standards as the basis for web*Mathematica* makes its integration into existing web infrastructures virtually seamless, not to mention future-proof," states Tom Wickham-Jones, Director of Kernel Development at Wolfram Research and chief developer of web*Mathematica*. "If you want to put interactive computations and visualizations on your website, there's no question that web*Mathematica* is the best and easiest way to do it."

web*Mathematica Japanese Edition* will make it easy and convenient for users to create entire interactive websites without making language an obstacle. It will ship with a fully localized Japanese version of *Mathematica*—including menus, palettes, dialog boxes, and error and warning messages—as well as full translations of the accompanying web*Mathematica* installation instructions, user guide, and other online and print documentation.

web*Mathematica* is initially available for Windows 95/98/Me/NT/2000, Intel-based Linux platforms, and Solaris. It comes in two license configurations: web*Mathematica Professional*, intended for development and deployment of sophisticated intranet and internet sites, and web*Mathematica Amateur*, designed for those who wish to showcase their work over the web. Subscribers to Wolfram Research's *Premier Service* can receive complimentary copies of web*Mathematica Amateur*.

Information on ordering web*Mathematica Japanese Edition* is available online. Visit the web*Mathematica* product pages for additional information, including working web*Mathematica* examples.

More info
www.wolfram.com/products/webmathematica

CalculationCenter 2

CalculationCenter 2 adds new calculation and communication capabilities.

CalculationCenter 2 from Wolfram Research, Inc. brings you new computing functionality, new communication capabilities, and further enhancements to this widely acclaimed calculation tool—all without compromising quality.

Initially released in March 2001, *CalculationCenter* has been highly rated by *Macworld*, *Desktop Engineering*, and *PC Plus*, among others, and was well received by users. *CalculationCenter* 2 builds upon the first version's innovative point-and-click interface concept, smart plotting, and unique InstantCalculators to provide a host of new features.

CalculationCenter 2 offers broad new statistics capabilities. The addition of analytical distribution functions for standard statistical distributions enables you to develop models that describe random behavior. Users can compare these models to experimental data to estimate parameters or can use them to generate random simulation data to test the likelihood of possible outcomes. Unlike typical statistics packages, *CalculationCenter* 2 works with the formulas related to statistical function, not just with the numbers, giving users much more insight and a greater range of problem-solving ability.

CalculationCenter's unique, intelligent error-correction technology has also been enhanced in Version 2. In the new version there are more cases in which *CalculationCenter* can save you time by suggesting what you meant to type. For users switching to *CalculationCenter* from Excel or Mathcad, there are even more features to help get you started in 10 minutes or less. Type or paste in a formula

containing an Excel or a Mathcad function, and *CalculationCenter* will suggest either the equivalent or the closest function available, updating the formula with the click of a button.

Other new capabilities in *CalculationCenter* 2 include:

- The addition of MathML 2.0 support, which allows users to better share their work over the web. MathML is the XML-based open standard for the use and re-use of mathematical and scientific content on the web.
- An extended range of graphic types with a new inequality plot that can visualize complicated linear and nonlinear, and continuous and discrete, inequality relationships. A new multiple-data-set plot type makes it easier than ever to compare data.
- Support for the 3D file formats DXF and STL and the data format CSV to facilitate interaction with other engineering, design, and data-handling software. There is also a new mechanism to import other *CalculationCenter* notebooks, making it easy to build up libraries of functions and to refer to them from within a notebook.
- New, faster algorithms for many of *CalculationCenter*'s data analysis functions. Now you can analyze and manipulate large data sets faster than ever before.

Containing hundreds of functions—most working with symbolic as well as numeric input and all with InstantCalculator support—*CalculationCenter* is the only tool needed to easily produce superior-quality technical reports that can be distributed via hard copy, email, or the web. Also ideal for organizations, *CalculationCenter* enables users of varying needs to share compatible documents and to maintain the notebook-document presentation style that is consistent with other Wolfram Research products.

CalculationCenter 2 is available for Windows 95/98/Me/NT/2000/XP and Mac OS platforms. The suggested retail price is $295 (United States and Canada), making it highly competitive with other calculating solutions. For international pricing, please contact your local reseller. For additional information about *CalculationCenter*, visit the Wolfram Research website.

More info
www.wolfram.com/products/calculationcenter

Mathematica Teacher's Edition

This toolkit is specially designed for math teachers to save them time, simplify their daily work, and help them create a positive classroom experience for their students.

Wolfram Research, Inc. is pleased to announce the release of the all new *Mathematica Teacher's Edition*. Designed exclusively for secondary-level math teachers, this software product combines powerful computation, time-saving tools, and

ready-to-use courseware with an intuitive user interface. *Mathematica Teacher's Edition* allows teachers to focus on the important aspects of teaching instead of on mundane, routine tasks.

Mathematica Teacher's Edition provides a wealth of tools to help teachers put their students on the fast track to learning. The unique point-and-click problem generator enables math educators to quickly and easily generate original assignments, quizzes, and answer keys. Instead of pulling predefined problems out of a database, *Mathematica Teacher's Edition* creates them on the fly—so teachers can compile new assignments each time, or even make them unique for each student.

The underlying *Mathematica* computational engine in *Mathematica Teacher's Edition* easily tackles everything from simple calculator problems to advanced symbolic manipulations, while the complete presentation environment helps teachers create their own classroom demonstrations and handouts. *Mathematica Teacher's Edition* also comes with built-in courseware and classroom demos, ranging from prealgebra to calculus, to enhance students' understanding and visualization capabilities. Teachers can use these resources directly or customize them for their particular curriculum.

Mathematica Teacher's Edition is the ideal tool to help teachers succeed in the classroom and bring mathematics alive for their students. With *Mathematica Teacher's Edition*, teachers can:

- Build customized assignments and quizzes in four easy steps
- Automatically generate answer keys and verify solutions
- Perform symbolic and numerical calculations in basic math, algebra, trigonometry, and calculus
- Create classroom demos featuring live 2D and 3D graphics, animations, and sounds—all designed to help augment students' intuition and interest
- Produce handouts with a full technical word-processing environment, including spell checking and mathematical typesetting
- Show students multiple problem-solving approaches to any given problem

Mathematica Teacher's Edition works seamlessly with *Mathematica for the Classroom* to give students a head start by exposing them to the computing tools and methods they will use throughout their academic and professional careers. Like all Wolfram Research products, *Mathematica Teacher's Edition* supports the platform-independent notebook (.nb) format. You can easily send notebooks by email or save them as HTML and post them on the web—making them a convenient way to share reports and documents between students and teachers.

Mathematica Teacher's Edition is available for Windows 95/98/Me/NT/2000/XP and Mac OS platforms. More information is available.

More info
www.wolfram.com/products/teachersedition

New *Mathematica* Applications

Analog Insydes 2.0

Release 2.0 of *Analog Insydes*, a *Mathematica* toolbox for the modeling, analysis, and design of analog electronic circuits, is now available.

More info
www.wolfram.com/products/applications/insydes

Digital Image Processing 1.1

The new version of *Digital Image Processing* provides a powerful collection of fundamental and advanced image processing tools.

More info
www.wolfram.com/products/applications/digitalimage

Global Optimization 4.2

Version 4.2 provides reliable global optimization for constrained and unconstrained nonlinear functions.

More info
www.wolfram.com/products/applications/globalopt

J/Link 2.0.1

J/Link integrates *Mathematica* and Java.

More info
www.wolfram.com/solutions/mathlink/jlink

Mathematica Link for LabVIEW 2.0

Mathematica Link for LabVIEW integrates design models and measurements.

More info
www.wolfram.com/products/applications/labview

MathOptimizer

MathOptimizer provides tools for solving a very general class of global and local optimization problems.

More info
www.wolfram.com/products/applications/mathoptimizer

mathStatica

mathStatica solves the algebraic and symbolic problems that are of primary interest in mathematical statistics.

More info
www.wolfram.com/products/applications/mathstatica

UnRisk PRICING ENGINE 1.6

UnRisk PRICING ENGINE is newly upgraded to bring even more speed and precision to the field of financial derivative analysis.

More info
www.wolfram.com/products/applications/unrisk

UnRisk STANDARD

UnRisk STANDARD provides standard utilities, functions, and derivative instruments for *Mathematica* programmers in finance.

More info
www.wolfram.com/products/applications/unriskstandard

New in the Library

SCI 2002 *Mathematica* Sessions

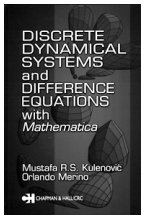
SCI 2002, the Sixth World Multiconference on Systemics, Cybernetics, and Informatics, was held from July 14 to 18, 2002, in Orlando, Florida. The conference included two sessions involving "Applications of *Mathematica* in Technology and Research." These sessions were organized by the Kernel Development group at Wolfram Research and included presentations from a variety of technical fields. Presentation materials are available for download at library.wolfram.com/papers/sci2002.

Resource Library How-Tos

A number of new *Mathematica* how-tos have been added to the library. New topics include:

- How Do I Perform CCD Image Reduction with *Mathematica* 4.2? library.wolfram.com/howtos/reduce
- How Do I Create *J/Link* Graphical User Interfaces in *Mathematica*? library.wolfram.com/howtos/jlinkgui
- How Do I Convert a *Mathematica* Notebook into a Web Application? library.wolfram.com/howtos/makemsp

New Publications



Discrete Dynamical Systems and Difference Equations with *Mathematica*

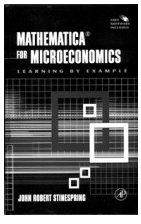
Mustafa R. S. Kulenovic and Orlando Merino, 2002, Chapman & Hall/CRC, 344 pp., hardcover.

This books combines an introductory survey of theory and techniques of discrete dynamical systems and difference equations with a manual for the use of the software package *Dynamica*. Many applications are presented as exercises and research projects. *Dynamica*, based on *Mathematica*, provides an easy-to-use collection of algebraic, numerical, and graphical tools and techniques that allow users to understand essential theoretical concepts quickly. The *Dynamica* package and notebooks corresponding to particular chapters are available for download from the internet.

Chapters

Dynamics of One-Dimensional Dynamical Systems | Dynamics of Two-Dimensional Dynamical Systems | Systems of Difference Equations, Stability, and Semicycles | Invariants and Related Lyapunov Functions | Dynamics of Three-Dimensional Dynamical Systems | Fractals Generated by Iterated Function Systems

Available:
store.wolfram.com/view/book/ISBN1584882875.str



Mathematica for Microeconomics: Learning by Example

John Robert Stinespring, 2002, Academic Press, 222 pp., hardcover, includes CD-ROM.

This book is intended for use as a supplemental tool for courses in advanced microeconomics and mathematical economics. Cogent applications of the mathematics required to solve problems of microeconomics are presented. An overview of *Mathematica* as well as tools to build individual *Mathematica* programs for problem solving are given. In the consumer theory section, common

problems of consumer optimization are explored. The producer theory section includes methods of transforming variables and manipulating output as well as linear algebra procedures. The section on economic dynamics utilizes the more advanced mathematical tools of differential and difference equations and the calculus of variations to study issues ranging from the dynamic adjustments of markets to the optimal amount of inventory held over time.

Chapters

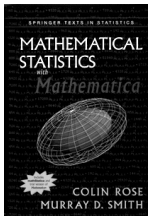
***Mathematica* Overview**: Introduction

Consumer Theory: Consumer Choice and the Lagrangian Multiplier Method | Individual and Market Demand | Compensating and Equivalent Variation | Pure Exchange | Intertemporal Trade | Choice under Uncertainty and Imperfect Information

Producer Theory: Cost Minimization | Short-Run and Long-Run Costs | Duality | Multiplant Production | Profit Maximization | Linear Programming | Production and Trade

Economic Dynamics: Market Dynamics | Dynamic Optimization and the Calculus of Variations

Available:
store.wolfram.com/view/book/ISBN0126709610.str



Mathematical Statistics with *Mathematica*

Colin Rose and Murray D. Smith, 2002, Springer,
481 pp., hardcover, includes two CD-ROMs.

This book and software package present a unified approach for doing mathematical statistics with *Mathematica*. The book, although it assumes preliminary knowledge of calculus, statistics, and probability, assumes no prior knowledge of *Mathematica*. The emphasis is on problem solving with less attention paid to the presentation of theorems and their associated proofs, which can be found in traditional texts.

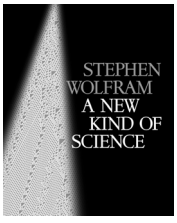
The *mathStatica* software, an add-on to *Mathematica*, provides a toolset specially designed for doing mathematical statistics. It enables students to solve difficult problems by removing the technical calculations often associated with mathematical statistics. The professional statistician will be able to tackle tricky multivariate distributions, generating functions, inversion theorems, symbolic maximum likelihood estimation, unbiased estimation, and the checking and correcting of textbook formulas. This text would be a useful companion for researchers and students in statistics, econometrics, engineering, physics, psychometrics, economics, finance, biometrics, and the social sciences.

The book contains two cross-platform CDs, which run on Windows, Mac OS, Linux, and most Unix platforms. The *mathStatica* CD includes *mathStatica*, the application package for mathematical statistics; custom *Mathematica* palettes; a live interactive book that is identical to the printed text; hundreds of live examples, animations, and illustrations; and online help. The second CD contains a trial version of *Mathematica* 4.

Chapters

Introduction | Continuous Random Variables | Discrete Random Variables | Distributions of Functions of Random Variables | Systems of Distributions | Multivariate Distributions | Moments of Sampling Distributions | Asymptotic Theory | Statistical Decision Theory | Unbiased Parameter Estimation | Principles of Maximum Likelihood Estimation | Maximum Likelihood Estimation in Practice | Appendix

Available:
store.wolfram.com/view/book/ISBN0387952349.str



A New Kind of Science

Stephen Wolfram, 2002, Wolfram Media, Inc.,
1197 pp., hardcover.

Two decades in the making, this long-awaited work from one of the world's most respected scientists presents a series of dramatic discoveries never before made public. Starting from a collection of simple computer experiments—illustrated in the book by striking computer graphics—Wolfram shows how their unexpected results force a whole new way of looking at the operation of our universe.

Wolfram uses his approach to tackle a remarkable array of fundamental problems in science: from the origin of the Second Law of Thermodynamics to the development of complexity in biology, the computational limitations of mathematics, the possibility of a truly fundamental theory of physics, and the interplay between free will and determinism.

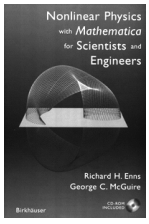
Written with exceptional clarity and illustrated by more than a thousand original pictures, this seminal book allows scientists and non-scientists alike to participate in what promises to be a major intellectual revolution.

Chapters

Preface | The Foundations for a New Kind of Science | The Crucial Experiment | The World of Simple Programs | Systems Based on Numbers | Two Dimensions and Beyond | Starting from Randomness | Mechanisms in Pro-

grams and Nature | Implications for Everyday Systems | Fundamental Physics | Processes of Perception and Analysis | The Notion of Computation | The Principle of Computational Equivalence | Notes | Index

Available:
store.wolfram.com/view/book/ISBN1579550088.str



Nonlinear Physics with *Mathematica* for Scientists and Engineers

Richard H. Enns and George C. McGuire, 2001, Birkhäuser, 691 pp., hardcover, includes CD-ROM.

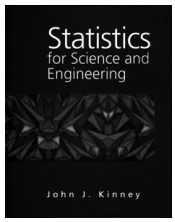
This text is aimed at a wide audience, including undergraduate and graduate students as well as working scientists. The experimental activities included are designed to deepen and broaden the reader's understanding of nonlinear physics. The *Mathematica* computer algebra system is used extensively, although no prior knowledge of *Mathematica* or programming is assumed. The CD-ROM included contains a wide variety of illustrative nonlinear examples solved with *Mathematica*. There are also 130 annotated *Mathematica* files that may be used to solve and explore the text's four hundred problems.

Chapters

Part I: Theory: Introduction | Nonlinear Systems, Part I | Nonlinear Systems, Part II | Topological Analysis | Analytic Methods | The Numerical Approach | Limit Cycles | Forced Oscillators | Nonlinear Maps | Nonlinear PDE Phenomena | Numerical Simulation | Inverse Scattering Method

Part II: Experimental Activities: Introduction to Nonlinear Experiments | Magnetic Force | Magnetic Tower | Spin Toy Pendulum | Driven Eardrum | Nonlinear Damping | Anharmonic Potential | Iron Core Inductor | Nonlinear LRC Circuit | Tunnel Diode Negative Resistance Curve | Tunnel Diode Self-Excited Oscillator | Forced Duffing Equation | Focal Point Instability | Compound Pendulum | Damped Simple Pendulum | Stable Limit Cycle | Van der Pol Limit Cycle | Relaxation Oscillations: Neon Bulb | Relaxation Oscillations: Drinking Bird | Relaxation Oscillations: Tunnel Diode | Hard Spring | Nonlinear Resonance Curve: Mechanical | Nonlinear Resonance Curve: Electrical | Nonlinear Resonance Curve: Magnetic | Subharmonic Response: Period Doubling | Diode: Period Doubling | Five-Well Magnetic Potential | Power Spectrum | Entrainment and Quasiperiodicity | Quasiperiodicity | Chua's Butterfly | Route to Chaos | Driven Spin Toy | Mapping

Available:
store.wolfram.com/view/book/ISBN0817642234.str



Statistics for Science and Engineering

John J. Kinney, 2002, Addison Wesley, 488 pp., hardcover.

This book is an introduction to the statistical analysis of data that arise from experiments, sample surveys, or other observational studies. The emphasis is on explaining statistical techniques and showing how they are used, not on discussing their mathematical origins. The book focuses on topics that are frequently used by scientists and engineers, particularly the topics of regression, design of experiments, and statistical process control. Exercises are drawn largely from applied situations in which scientists and engineers are likely to find themselves. *Mathematica* is used to illustrate results graphically and to do complex calculations, and Appendix A provides instruction and guidance in the use of *Mathematica*.

Chapters

Graphs and Statistics | Random Variables and Probability Distributions | Estimation and Hypothesis Testing | Simple Linear Regression—Summarizing Data with Equations | Multiple Linear Regression | Design of Science and Engineering Experiments | Statistical Process Control

Available:
store.wolfram.com/view/book/ISBN0201437201.str

VisualDSolve

Stan Wagon and Dan Schwalbe, downloadable *Mathematica* application.

VisualDSolve is a *Mathematica* package and accompanying e-book showing how *Mathematica*'s visualization tools can be used to enhance the viewing of solutions to differential equations. The book contains 17 chapters; the first five serve as a manual to the many functions of the package, and the other 12 are comprehensive examples. The package itself has some new ways of viewing differential equations. Two noteworthy examples are (1) the use of curvy fish shapes to describe the vector fields determined by a differential equation and (2) the use of colored nullcline regions to describe the different types of directions for an autonomous system of two equations.

Chapters

VisualDSolve | Auxiliary Functions | SystemSolutionPlot | PhasePlot | Second-OrderPlot | Differential Equations and *Mathematica* | Some Parachute Experiments | Logistic Models of Population Growth | Linear Systems | Hamiltonian Systems | A Devilish Equation | Lead Flow in the Human Body | Making a Discus Fly | Double Pendulum | The Duffing Equation | The Tetrapods of Wada | Rigid Bodies | Appendices

Available:
store.wolfram.com/view/book/D0706.str

Tricks of the Trade

Edited by Paul Abbott

Department of Physics
University of Western Australia
35 Stirling Highway
Nedlands WA 6009, Australia
tmj@physics.uwa.edu.au
physics.uwa.edu.au/~paul

This is a column of programming tricks and techniques, most of which, we hope, will be contributed by our readers.

■ Inflight Conundrums

Here are two puzzles taken from an inflight magazine.

□ Puzzle 1

Which symbol should replace the ⚠?

✸	□	✸	□	26
□	○	○	✸	
○	✸	○	□	
○	□	⚠	⚠	29
23	20			

First, express this table of symbols as a matrix.

```
In[1]:= M = ( ✸ □ ✸ □
              □ ○ ○ ✸
              ○ ✸ ○ □
              ○ □ ⚠ ⚠ );
```

Then, introduce the unknowns a and b to represent the sum of the second and third rows respectively, so as to obtain equations for each row.

In[2]:= **rowsums = $\mathcal{M}$.** $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ **==** $\begin{pmatrix} 26 \\ a \\ b \\ 29 \end{pmatrix}$

Out[2]= $\begin{pmatrix} 2\square + 2\star \\ 2\bigcirc + \square + \star \\ 2\bigcirc + \square + \star \\ \bigcirc + \square + 2\triangle \end{pmatrix} == \begin{pmatrix} 26 \\ a \\ b \\ 29 \end{pmatrix}$

Similarly, introduce c and d for the sum of the third and fourth columns respectively.

In[3]:= **colsums = (1 1 1 1).$\mathcal{M}$ == (23 20 *c* *d*)**

Out[3]= $(2\bigcirc + \square + \star \quad \bigcirc + 2\square + \star \quad 2\bigcirc + \star + \triangle \quad 2\square + \star + \triangle) == (23 \quad 20 \quad c \quad d)$

Together, these equations allow us to determine all the unknowns.

In[4]:= **Solve[{rowsums, colsums}, Join[Variables[$\mathcal{M}$], {*a*, *b*, *c*, *d*}]] // First**

Out[4]= $\{\bigcirc \to 5, \square \to 2, \star \to 11, \triangle \to 11, a \to 23, b \to 23, c \to 32, d \to 26\}$

To determine which symbol should replace the $\triangle$, we use pattern-matching.

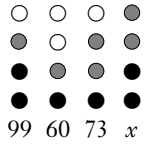
In[5]:= **Cases[% /. (x_ → (△ /. %)) → *x*, _Symbol]**

Out[5]= {★, △}

Hence ★ should replace △.

□ Puzzle 2

What is the value of the column with total x?



The solution is immediate.

In[1]:= **Solve[(1 1 1 1).** $\begin{pmatrix} \bigcirc & \bigcirc & \bigcirc & \bullet \\ \bullet & \bigcirc & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}$ **== (99 60 73 *x*), *x*]**

Out[1]= $\{\{x \to 112\}\}$

Indeed, you can solve for all symbols if you want.

```
In[2]:= Solve[(1 1 1 1).{{○, ○, ○, ●}, {●, ○, ●, ●}, {●, ●, ●, ●}} == (99 60 73 x), {x, ○, ●, ●}]
Out[2]= {{x → 112, ○ → 2, ● → 15, ● → 41}}
```

■ Integral Equations

Inhomogeneous Fredholm integral equations of the second kind,

$$f(x) = g(x) + \lambda \int_a^b \kappa(x, t)\, f(t)\, dt, \quad a \le x \le b, \tag{1}$$

involve solving for $f(x)$, where $\kappa(x, t)$ is the integral kernel, λ is a parameter, and $g(x)$ is the inhomogeneous term [1, 2]. *Mathematica* does not include built-in routines or packages for integral equations. However, it is straightforward to implement quite general methods for solving such equations.

As a specific example, consider Love's equation

$$f(x) = 1 + \frac{1}{\pi} \int_{-1}^{1} \frac{f(t)}{(x-t)^2 + 1}\, dt, \tag{2}$$

which arises in electrostatics [3, 4]. An excellent approximation to the solution $f(x)$ can be found in the form of a *Chebyshev-series*

$$f(x) = \sum_{i=0}^{n} c_i\, T_{2i}(x). \tag{3}$$

The coefficients c_i for $i = 0, 1, \ldots, n$, are determined by solving numerically the simultaneous linear algebraic equations that result by evaluating the integral equation (2) at the *Chebyshev points* $x_j = \cos\left(\frac{j\pi}{2n}\right)$ for $j = 0, 1, \ldots, 2n$, using the Chebyshev-series (3) [5, 6].

Here is a direct implementation for $n = 4$. The Chebyshev points are calculated using arbitrary-precision arithmetic.

```
In[1]:= n = 4; xs = N[cos(Range[0, 2 n] π / (2 n)), 20]
Out[1]= {1.0000000000000000000, 0.92387953251128675613,
  0.70710678118654752440, 0.38268343236508977173, 0,
  -0.38268343236508977173, -0.70710678118654752440,
  -0.92387953251128675613, -1.0000000000000000000}
```

Instead of computing the points using **Table**, we have used the **Listable** attribute of **Cos**.

We introduce the undetermined coefficients c_i.

In[2]:= **cs = Thread[$c_{\text{Range}[0,n]}$]**

Out[2]= $\{c_0, c_1, c_2, c_3, c_4\}$

We evaluate the left-hand side of equation (2) at the Chebyshev points.

In[3]:= **lhs = cs.Table[T_{2i}(xs), {i, 0, n}];**

Note that we have replaced the sum in the Chebyshev-series (3) by a dot product and have used the **Listable** attribute of **ChebyshevT** to compute the value at *all* the Chebyshev points at once.

In[4]:= **Attributes[ChebyshevT]**

Out[4]= {Listable, NumericFunction, Protected, ReadProtected}

To compute the right-hand side of equation (2) we could use a Chebyshev quadrature formula [5, 6]. Instead we use **NIntegrate** for simplicity, again using arbitrary-precision arithmetic.

In[5]:= **rhs = 1 + $\frac{1}{\pi}$ cs.Table[NIntegrate[Evaluate[$\frac{T_{2i}(t)}{(\text{xs}-t)^2+1}$],**
{t, −1, 1}, WorkingPrecision → 20], {i, 0, n}];

Solving the resulting simultaneous linear algebraic equations we obtain the coefficients c_i for $i = 0, 1, \ldots, n$.

In[6]:= **Solve[lhs == rhs, cs]**

Out[6]= $\{\{c_0 \to 1.774444805, c_1 \to -0.1400430745, c_2 \to 0.004962498177,$
$c_3 \to 0.0003746723341, c_4 \to -0.00004368650886\}\}$

The solution $f(x)$ is obtained from the Chebyshev-series (3).

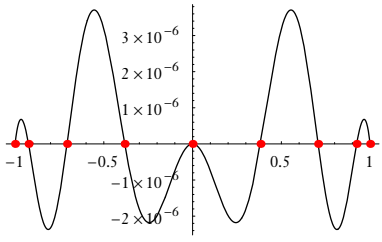
In[7]:= **f(x_) = $\sum_{i=0}^{n} c_i\, T_{2i}(x)$ /. First[%]**

Out[7]= $-0.1400430745\,(2x^2-1) + 0.004962498177\,(8x^4-8x^2+1) +$
$0.0003746723341\,(32x^6-48x^4+18x^2-1) -$
$0.00004368650886\,(128x^8-256x^6+160x^4-32x^2+1) + 1.774444805$

To see how well the Chebyshev-series solution satisfies Love's equation (2), we substitute it back into the equation and plot the difference of the left- and right-hand sides. The Chebyshev points are also displayed and it is apparent that the *minimum error* in the Chebyshev-series solution occurs at these points.

```
In[8]:= Plot[1 + 1/π NIntegrate[f(t)/((x - t)^2 + 1), {t, -1, 1}] - f(x),
          {x, -1, 1}, Epilog → {Hue[0], AbsolutePointSize[3],
            Point /@ (xs /. x_ ?NumberQ → {x, 0})}, PlotRange → All];
```

From In[8]:=



As an aside, we can get a bound for the *maximum error* directly from the **PlotRange** of this plot. We could use this error-bound in a recursive procedure so as to find a solution for a given maximum error.

```
In[9]:= Max[Abs[Last[PlotRange[%]]]]
```

$$\text{Out[9]=}\quad 3.85916 \times 10^{-6}$$

■ Matrix Differential Operator

The curl is often represented as a differential operator.

```
In[1]:= ∇ /: ∇×{f_, g_, h_} := {∂h/∂y - ∂g/∂z, ∂f/∂z - ∂h/∂x, ∂g/∂x - ∂f/∂y}
```

Here is an example.

```
In[2]:= ∇×{f(y), g(z), h(x)}
```

$$\text{Out[2]=}\quad \{-g'(z), -h'(x), -f'(y)\}$$

Alternatively, curl can be represented as a matrix differential operator, C.

$$C = \begin{pmatrix} 0 & -\partial_z & \partial_y \\ \partial_z & 0 & -\partial_x \\ -\partial_y & \partial_x & 0 \end{pmatrix}.$$

Using pure function notation (see Section 2.2.5 of *The Mathematica Book*) this reads as follows.

```
In[3]:= C := {{0 &, -∂z # &, ∂y # &},
              {∂z # &, 0 &, -∂x # &},
              {-∂y # &, ∂x # &, 0 &}}
```

However, we cannot **Dot** this operator by a vector to compute the curl because each element resulting from the dot product operation must be *composed* instead of being multiplied. **Inner** generalizes **Dot** and allows us to do what we want. Here we use the **CenterDot** operator (see Section 3.10.4 of *The Mathematica Book*) to define a suitable shorthand notation.

```
In[4]:= a_ · b_ := Inner[#1[#2] &, a, b]
```

Now we can compute the curl using the matrix differential operator.

```
In[5]:= C · {f(y), g(z), h(x)}
```

$$\text{Out[5]=}\quad \{-g'(z),\ -h'(x),\ -f'(y)\}$$

As a second example, Kurasov and Naboko [7] study the essential spectrum of the following matrix differential operator:

$$\mathcal{D} = \begin{pmatrix} -\dfrac{d}{dx}\,\rho(x)\,\dfrac{d}{dx} + q(x) & \dfrac{\beta(x)}{x}\,\dfrac{d}{dx} \\ -\dfrac{\beta(x)}{x}\,\dfrac{d}{dx} & \dfrac{m(x)}{x^2} \end{pmatrix}. \tag{4}$$

The four differential operators in the matrix $\mathcal{D}$ can be implemented as follows.

```
In[6]:= 𝒟1,1[f_] := -∂x (ρ(x) ∂f/∂x) + q(x) f
        𝒟1,2[f_] := ∂x (β(x)/x f)
        𝒟2,1[f_] := -(β(x)/x) ∂f/∂x
        𝒟2,2[f_] := m(x)/x^2 f
```

Let us modify the action of the **CenterDot** operator by applying **HoldForm** to the first argument, so that it does not evaluate the operators.

```
In[10]:= a_ · b_ := Inner[HoldForm[#1][#2] &, a, b]
```

We now can examine the application of the matrix differential operator to a vector or matrix before we explicitly evaluate the matrix entries using **ReleaseHold**.

```
In[11]:= Table[𝒟i,j, {i, 2}, {j, 2}] · (f(x)
                                        g(x))
```

$$\text{Out[11]=}\quad \begin{pmatrix} \mathcal{D}_{1,1}[f(x)] + \mathcal{D}_{1,2}[g(x)] \\ \mathcal{D}_{2,1}[f(x)] + \mathcal{D}_{2,2}[g(x)] \end{pmatrix}$$

```
In[12]:= ReleaseHold[%]
```

$$\text{Out[12]=}\quad \begin{pmatrix} f(x)\,q(x) - \frac{g(x)\,\beta(x)}{x^2} + \frac{\beta(x)\,g'(x)}{x} + \frac{g(x)\,\beta'(x)}{x} - f'(x)\,\rho'(x) - \rho(x)\,f''(x) \\ \frac{g(x)\,m(x)}{x^2} - \frac{\beta(x)\,f'(x)}{x} \end{pmatrix}$$

As an aside, if we use the **SmallCircle** operator to denote **Composition**, then formal products of differential operators can be computed quite elegantly.

```
In[13]:= SmallCircle = Composition;
```

For example, the action of the product of the antidiagonal elements in $\mathcal{D}$ can be computed as follows.

```
In[14]:= (D[1,2] ∘ D[2,1])[f(x)]
```

$$\text{Out[14]=}\quad \frac{2\,f'(x)\,\beta(x)^2}{x^3} - \frac{f''(x)\,\beta(x)^2}{x^2} - \frac{2\,f'(x)\,\beta'(x)\,\beta(x)}{x^2}$$

References

[1] C. T. H. Baker, *The Numerical Treatment of Integral Equations*, Oxford: Clarendon Press, 1977 pp. 358–60.

[2] C. E. Pearson, *Handbook of Applied Mathematics*, New York: Van Nostrand Reinhold, 1990.

[3] E. R. Love, "The Electrostatic Field of Two Equal Circular Co-axial Conducting Disks," *Quarterly Journal of Mechanics and Applied Mathematics*, **2**, 1949 pp. 428–451.

[4] L. Fox and E. T. Goodwin, "The Numerical Solution of Non-singular Linear Integral Equations," *Philosophical Transactions of the Royal Society of London*, Series A, **245**, 1953 pp. 501–534.

[5] C. W. Clenshaw and A. R. Curtis, "A Method for Numerical Integration on an Automatic Computer," *Numerische Mathematik*, **2**, 1960 pp. 197–205.

[6] S. E. El-gendi, "Chebyshev Solution of Differential, Integral and Integro-differential Equations," *The Computer Journal*, **12**, 1969 pp. 282–287. http://www3.oup.co.uk/computer_journal/hdb/Volume_12/Issue_03.

[7] P. Kurasov and S. Naboko, "On the Essential Spectrum of a Class of Singular Matrix Differential Operators. I Quasiregularity Conditions and Essential Self-adjointness," *Research Reports in Mathematics, Department of Mathematics, Stockholm University*, **12**, 2001. http://www.matematik.su.se/reports/2001/12.

In and Out

Edited by Paul Abbott

Department of Physics
University of Western Australia
35 Stirling Highway
Nedlands WA 6009, Australia
tmj@physics.uwa.edu.au
physics.uwa.edu.au/~paul

In and Out offers readers an opportunity to ask questions of the experts. *The Mathematica Journal* encourages readers to submit problems in care of the editor.

■ Integrals over Chebyshev Polynomials

Q: How can I compute

$$\mathcal{I}_{nmq} = \int_{-1}^{1} T_n(x) \frac{\partial T_m(x)}{\partial x} \frac{\partial T_q(x)}{\partial x} dx$$

for general non-negative integers n, m, and q, where $T_n(x)$ is the **ChebyshevT** function?

A: *Mathematica* cannot compute the general integral directly. However, the integrand simplifies to the following.

```
In[1]:= T_n(x) ∂T_m(x)/∂x ∂T_q(x)/∂x
Out[1]= m q T_n(x) U_(m-1)(x) U_(q-1)(x)
```

Here $U_n(x)$ is the **ChebyshevU** function (see Section 3.2.9 of *The Mathematica Book*). These orthogonal polynomials have very simple definitions in terms of trigonometric functions:

$$T_n(\cos(\theta)) = \cos(n\theta),$$

$$U_n(\cos(\theta)) = \frac{\sin((n+1)\theta)}{\sin(\theta)},$$

which suggests the change of variables $x \to \cos(\theta)$, $dx \to -\sin(\theta)\, d\theta$.

Moreover, there is a very simple *addition theorem* for the product of two **ChebyshevU** functions,

$$U_m(x)\,U_n(x) = U_{|m-n|}(x) + U_{|m-n|+2}(x) + \cdots + U_{m+n-2}(x) + U_{m+n}(x).$$

This result can be proved using *Mathematica* as follows. Assume without loss of generality that $n = k + m$. Then we want to show that

$$U_m(x)\,U_{k+m}(x) = U_k(x) + U_{k+2}(x) + \cdots + U_{k+2\,m-2}(x) + U_{k+2\,m}(x) = \sum_{p=\frac{k}{2}}^{\frac{k}{2}+m} U_{2\,p}(x).$$

Using the trigonometric definition for $U_n(\cos(\theta))$, this sum can be computed explicitly and simplified using **ComplexExpand**.

```
In[2]:= Simplify[Sum[Sin[(2 p + 1) θ]/Sin[θ], {p, k/2, k/2 + m}], m ∈ Z]
```

$$\text{Out[2]=}\quad \frac{e^{-i(k+2m)\theta}\,(-1+e^{2i(m+1)\theta})\,(-1+e^{2i(k+m+1)\theta})}{(-1+e^{2i\theta})^2}$$

```
In[3]:= ComplexExpand[%, TargetFunctions → {Re, Im}] // Simplify
```

$$\text{Out[3]=}\quad \csc^2(\theta)\sin((m+1)\theta)\sin((k+m+1)\theta)$$

It is clear that this result is just $U_m(\cos(\theta))\,U_{k+m}(\cos(\theta))$.

Now we are in a position to compute the given integral in closed form:

$$\int_{-1}^{1} T_n(x)\,\frac{\partial T_m(x)}{\partial x}\,\frac{\partial T_q(x)}{\partial x}\,dx$$

$$= m\,q\int_{-1}^{1} T_n(x)\,U_{m-1}(x)\,U_{q-1}(x)\,dx \qquad \text{differentiation of } T_n$$

$$= m\,q\int_{0}^{\pi} T_n(\cos(\theta))\,U_{m-1}(\cos(\theta))\,U_{q-1}(\cos(\theta))\sin(\theta)\,d\theta \qquad x \to \cos(\theta)$$

$$= m\,q\sum_{p=\frac{|q-m|}{2}}^{\frac{q+m-2}{2}}\int_{0}^{\pi} T_n(\cos(\theta))\,U_{2\,p}(\cos(\theta))\sin(\theta)\,d\theta \qquad \text{addition theorem}$$

$$= m\,q\sum_{p=\frac{|q-m|}{2}}^{\frac{q+m-2}{2}}\int_{0}^{\pi} \cos(n\,\theta)\,\frac{\sin((2\,p+1)\,\theta)}{\sin(\theta)}\sin(\theta)\,d\theta. \qquad \text{definition of } T_n \text{ and } U_n$$

The integral in the last line can be computed and simplified by applying the conditions that $n \in \mathbb{Z}$ and $2\,p \in \mathbb{Z}$.

```
In[4]:= Simplify[∫_0^π Cos[n θ] Sin[(2 p + 1) θ]/Sin[θ] Sin[θ] dθ, {n, 2 p} ∈ ℤ]
```

$$\text{Out[4]=}\quad -\frac{(1+(-1)^{n+2p})(2p+1)}{(n-2p-1)(n+2p+1)}$$

Note that although $(n-2p-1)$ appears in the denominator of this expression, there is no problem if $n=2p+1$ because the integral vanishes in that case.

```
In[5]:= Simplify[∫_0^π Cos[(2 p + 1) θ] Sin[(2 p + 1) θ]/Sin[θ] Sin[θ] dθ, 2 p ∈ ℤ]
```

$$\text{Out[5]=}\quad 0$$

More generally, the integral vanishes if $n+2p$ is odd, and the summation limits ($2p$ ranges from $|q-m|$ to $q+m-2$ in steps of 2) imply that the original integral vanishes unless $n+m+q$ is even. This restriction results from the parity (i.e., whether the function is even or odd) of the Chebyshev polynomials. For example, with $n=2$, $m=3$, and $q=6$, the resulting polynomial is odd with respect to the interchange $x \leftrightarrow -x$, and hence the integral of this quantity over $x \in [-1, 1]$ will vanish identically.

```
In[6]:= T_2[x] ∂T_3[x]/∂x ∂T_6[x]/∂x // Simplify
```

$$\text{Out[6]=}\quad 36x(2x^2-1)(4x^2-1)(16x^4-16x^2+3)$$

Hence, for $n+m+q$ even, the summand can be written as

$$\int_{-1}^{1} T_n(x)\frac{\partial T_m(x)}{\partial x}\frac{\partial T_q(x)}{\partial x}dx = -2mq\sum_{p=\frac{|q-m|}{2}}^{\frac{q+m-2}{2}}\frac{2p+1}{(n-2p-1)(n+2p+1)}\,.$$

We see that both sides of this equation are symmetric with respect to the interchange $m \leftrightarrow q$, as expected, which also means that we can choose the labels m and q so that $q-m \geq 0$.

Mathematica implementation is immediate.

```
In[7]:= S[n_Integer, m_Integer, q_Integer] :=
          -2 m q Sum[(2 p + 1)/((n - 2 p - 1) (n + 2 p + 1)), {p, Abs[q - m]/2, (q + m - 2)/2}] /; EvenQ[m + n + q]
```

We also enter this "catch-all" rule.

```
In[8]:= S[n_, m_, q_] = 0;
```

Note that *Mathematica* can compute this sum in closed form as follows.

```
In[9]:= Simplify[-2 m q Sum[(2 p + 1)/((n - 2 p - 1) (n + 2 p + 1)), {p, |q - m|/2, (q + m - 2)/2}]]
```

$$\text{Out[9]}= \frac{1}{2}\, m\, q \left(\psi^{(0)}\left(\frac{1}{2}\,(m-n+q+1)\right)+\psi^{(0)}\left(\frac{1}{2}\,(m+n+q+1)\right)- \psi^{(0)}\left(\frac{1}{2}\,(-n+|m-q|+1)\right)-\psi^{(0)}\left(\frac{1}{2}\,(n+|m-q|+1)\right)\right)$$

So, if required, we also have a relatively simple expression for the original integral.

Entering the original integral as a check, we put $n = 17$, $m = 12$, and $q = 37$ and compare this with the summation result.

```
In[10]:= I[n_Integer, m_Integer, q_Integer] := Integrate[ChebyshevT[n, x] D[ChebyshevT[m, x], x] D[ChebyshevT[q, x], x], {x, -1, 1}]

In[11]:= I[17, 12, 37]

Out[11]= 1654280971704837341102624/4059090323908283438775

In[12]:= S[17, 12, 37]

Out[12]= 1654280971704837341102624/4059090323908283438775
```

Next we check all cases $0 \le n \le 6$, $0 \le m \le 6$, and $0 \le q \le m$ (we can truncate this last range because of the symmetry under the interchange $m \leftrightarrow q$).

```
In[13]:= And @@ Flatten[Table[I[n, m, q] == S[n, m, q], {n, 0, 6}, {m, 0, 6}, {q, 0, m}]]

Out[13]= True
```

An alternate approach involves using the generating functions,

$$\frac{1 - x\,z}{1 - 2\,x\,z + z^2} = \sum_{n=0}^{\infty} z^n\, T_n(x),$$

and

$$\frac{1}{1 - 2\,x\,z + z^2} = \sum_{n=0}^{\infty} z^n\, U_n(x),$$

valid for $|z| < 1$. Formally, we have

$$\int_{-1}^{1} \frac{1 - \alpha\,x}{(1 - 2\,\alpha\,x + \alpha^2)\,(1 - 2\,\beta\,x + \beta^2)\,(1 - 2\,\gamma\,x + \gamma^2)}\, dx = \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} \sum_{q=1}^{\infty} \alpha^n\, \beta^{m-1}\, \gamma^{q-1} \left(\int_{-1}^{1} T_n(x)\, U_{m-1}(x)\, U_{q-1}(x)\, dx \right).$$

We now compute and simplify the integral on the right-hand side.

```
In[14]:= Collect[Simplify[∫_-1^1 (1 - α x)/((1 - 2 α x + α^2) (1 - 2 β x + β^2) (1 - 2 γ x + γ^2)) dx,
           -1 < α < 1 ∧ -1 < β < 1 ∧ -1 < γ < 1], {log(_)}, Factor]
```

$$\begin{aligned}\text{Out[14]}= &\frac{(\alpha-1)\,\alpha\,(\alpha+1)\log(-(\alpha-1)^2)}{4\,(\alpha-\beta)\,(\alpha\,\beta-1)\,(\alpha-\gamma)\,(\alpha\,\gamma-1)} - \frac{(\alpha-1)\,\alpha\,(\alpha+1)\log(-(\alpha+1)^2)}{4\,(\alpha-\beta)\,(\alpha\,\beta-1)\,(\alpha-\gamma)\,(\alpha\,\gamma-1)} + \\ &\frac{(\alpha\,\beta^2-2\,\beta+\alpha)\log((\beta-1)^2)}{4\,(\beta-\alpha)\,(\alpha\,\beta-1)\,(\beta-\gamma)\,(\beta\,\gamma-1)} - \frac{(\alpha\,\beta^2-2\,\beta+\alpha)\log(\beta+1)}{2\,(\beta-\alpha)\,(\alpha\,\beta-1)\,(\beta-\gamma)\,(\beta\,\gamma-1)} + \\ &\frac{(\alpha\,\gamma^2-2\,\gamma+\alpha)\log((\gamma-1)^2)}{4\,(\alpha-\gamma)\,(\beta-\gamma)\,(\alpha\,\gamma-1)\,(\beta\,\gamma-1)} - \frac{(\alpha\,\gamma^2-2\,\gamma+\alpha)\log(\gamma+1)}{2\,(\alpha-\gamma)\,(\beta-\gamma)\,(\alpha\,\gamma-1)\,(\beta\,\gamma-1)}\end{aligned}$$

Although valid, this result can be further simplified. For example, most readers would prefer to see $2\log\left(\frac{1-\alpha}{1+\alpha}\right)$ for $\alpha \in [-1, 1]$ instead of $\log\left(-(\alpha-1)^2\right) - \log\left(-(\alpha+1)^2\right)$ since it is straightforward to show that these expressions are equivalent.

```
In[15]:= FullSimplify[log(-(α - 1)^2) - log(-(α + 1)^2) - 2 log((1 - α)/(1 + α)), -1 < α < 1]
```

```
Out[15]= 0
```

However, there does not seem to be a straightforward way of forcing this simplification.

Series expansion generates the required integrals.

```
In[16]:= Series[%%, {α, 0, 2}, {β, 0, 2}, {γ, 0, 2}] // Normal // Expand
```

$$\begin{aligned}\text{Out[16]}= &\frac{6\,\beta^2\,\alpha^2}{5} + \frac{106}{105}\,\beta^2\,\gamma^2\,\alpha^2 + \frac{6\,\gamma^2\,\alpha^2}{5} + \frac{8}{15}\,\beta\,\gamma\,\alpha^2 - \frac{2\,\alpha^2}{3} + \frac{28}{15}\,\beta\,\gamma^2\,\alpha + \\ &\frac{4\,\beta\,\alpha}{3} + \frac{28}{15}\,\beta^2\,\gamma\,\alpha + \frac{4\,\gamma\,\alpha}{3} + \frac{2\,\beta^2}{3} + \frac{46\,\beta^2\,\gamma^2}{15} + \frac{2\,\gamma^2}{3} + \frac{8\,\beta\,\gamma}{3} + 2\end{aligned}$$

As a check, we compute the left-hand side up to the same order using $\mathcal{I}[n, m, q]$ or $\mathcal{S}[n, m, q]$.

```
In[17]:= % == Sum[Sum[Sum[α^n β^(m-1) γ^(q-1) 1/(m q) ℐ[n, m, q], {q, 1, 3}], {m, 1, 3}], {n, 0, 2}]
```

```
Out[17]= True
```

```
In[18]:= %% == Sum[Sum[Sum[α^n β^(m-1) γ^(q-1) 1/(m q) 𝒮[n, m, q], {q, 1, 3}], {m, 1, 3}], {n, 0, 2}]
```

```
Out[18]= True
```

■ Times → CircleTimes

Q: Is it possible to have *Mathematica* interpret space *not* as multiplication but instead as noncommutative multiplication?

A: Ordinarily *Mathematica* interprets space as (commutative) multiplication.

```
In[1]:= b a
Out[1]= a b
```

It is possible to have the implied multiplication interpreted as noncommutative multiplication. One way to do this is to use **$Pre** to replace **Times** by any operator that is not, by default, commutative. For example, here the **CircleTimes** operator (⊗) is used.

```
In[2]:= $Pre :=
          Function[x, ReleaseHold[Hold[x] /. Times → CircleTimes], HoldAll]
```

Note that we have to hold the argument (**Hold[x]**), and that the pure function needs to have the **HoldAll** attribute so that its argument is not evaluated.

Now $b\,a$ is automatically parsed to $b \otimes a$.

```
In[3]:= b a
Out[3]= b ⊗ a
```

Scalars appearing in expressions are not presently handled automatically.

```
In[4]:= 2 a 3 b
Out[4]= 2 ⊗ a ⊗ 3 ⊗ b
```

Here we extend the definition of **$Pre** to handle scalars.

```
In[5]:= Clear[$Pre];
        $Pre := Function[x, ReleaseHold[Hold[x] /. Times → CircleTimes //.
                  {a___ ⊗ b_ ? NumericQ ⊗ c___ → b a ⊗ c, ⊗a_ → a}], HoldAll]
```

```
In[7]:= 2 a 3 b
Out[7]= 6 a ⊗ b
```

Specific properties of a particular noncommutative operator can be added as definitions to the (infix) **CircleTimes** operator. For example, after adding the commutation relation $b \otimes a = -a \otimes b$, where variables are sorted lexicographically, sums of noncommutative products are ordered and simplified.

```
In[8]:= CircleTimes /: a_Symbol ⊗ b__Symbol :=
          Signature[{a, b}] CircleTimes @@ Sort[{a, b}] /; ¬ OrderedQ[{a, b}]
```

```
In[9]:= 2 a b + b a
Out[9]= a ⊗ b
```

It is easy to check that the sum of the noncommutative product of all the permutations of a set of symbols is zero for this commutation relation.

```
In[10]:= Tr[CircleTimes @@@ Permutations[{a, b, c, d, e}]]
Out[10]= 0
```

If we clear the values of **$Pre** and **CircleTimes** the interpretation of a space as ordinary multiplication is restored.

```
In[11]:= Clear[$Pre, CircleTimes]
In[12]:= b a
Out[12]= a b
```

■ Matrix Derivative

Q: Given that $A(t)$ is a $m \times n$ matrix that varies with time t, define its Moore–Penrose pseudoinverse, $A^+(t) = A(t)^T.\left(A(t).A(t)^T\right)^{-1}$. How can I find a simplified expression for $\frac{\partial(A^+(t))}{\partial t}$ in terms of $A(t)$?

A: *Mathematica* does not know how to differentiate an arbitrary matrix object.

```
In[1]:= ∂M(t)^-1/∂t
Out[1]= Inverse'[M(t)] M'(t)
```

Noting that

$$M(t).M(t)^{-1} = I$$

$$\Rightarrow \frac{\partial\left(M(t).M(t)^{-1}\right)}{\partial t} = M'(t).M(t)^{-1} + M(t).\frac{\partial M(t)^{-1}}{\partial t} = 0$$

$$\Rightarrow \frac{\partial M(t)^{-1}}{\partial t} = -M(t)^{-1}.M'(t).M(t)^{-1},$$

one approach is to define a rule that *formally* computes the derivative of the **Inverse** of an arbitrary invertible matrix, dividing through by the "additional" $\frac{\partial M}{\partial t}$ generated by the chain rule.

```
In[2]:= Derivative /: Inverse'[M_] := -M^-1 . ∂M/∂t . M^-1 / ∂M/∂t;

        ∂M(t)^-1
        --------
          ∂t

Out[3]= -M(t)^-1 . M'(t) . M(t)^-1
```

We also see that *Mathematica* does not know how to differentiate the transpose of a matrix object.

```
In[4]:= ∂M(t)^T
        -------
          ∂t

Out[4]= M'(t) Transpose'[M(t)]
```

So we define a rule to correctly differentiate **Transpose**, again dividing through by $\frac{\partial M}{\partial t}$.

```
In[5]:= Derivative /: Transpose'[M_] := (∂M/∂t)^T / ∂M/∂t;

        ∂M(t)^T
        -------
          ∂t

Out[6]= M'(t)^T
```

Introducing the Moore–Penrose pseudoinverse, $A^+(t) = A(t)^T.\left(A(t).A(t)^T\right)^{-1}$, as follows, we can now compute its derivative.

```
In[7]:= B(t_) := (A(t).A(t)^T)^-1

In[8]:= A^+(t_) := A(t)^T . B(t)

In[9]:= ∂A^+(t)
        -------
          ∂t

Out[9]= A(t)^T.(-(A(t).A(t)^T)^-1.(A(t).A'(t)^T + A'(t).A(t)^T).(A(t).A(t)^T)^-1) +
          A'(t)^T.(A(t).A(t)^T)^-1
```

To expand this result we add some rules to **Dot**.

```
In[10]:= Unprotect[Dot]; a___.(b_?NumericQ c_).d___ := b a.c.d;
           a_.(b_ + c_) := a.b + a.c; (b_ + c_).d_ := b.d + c.d; Protect[Dot];

In[11]:= ∂A^+(t)
         -------
           ∂t

Out[11]= A'(t)^T.(A(t).A(t)^T)^-1 - A(t)^T.(A(t).A(t)^T)^-1.A(t).A'(t)^T.(A(t).A(t)^T)^-1 -
           A(t)^T.(A(t).A(t)^T)^-1.A'(t).A(t)^T.(A(t).A(t)^T)^-1
```

This expression simplifies somewhat, noting that $A^+(t)$ and $B(t)$ appear in the result.

```
In[12]:= simplified = % //. {B(t) → HoldForm [B(t)], A⁺(t) → HoldForm [A⁺(t)]}
```

$$\text{Out[12]}= A'(t)^T.B(t) - A^+(t).A'(t).A^+(t) - A^+(t).A(t).A'(t)^T.B(t)$$

We have used **HoldForm** so that we can recover the explicit expression if required.

```
In[13]:= ReleaseHold[simplified]
```

$$\text{Out[13]}= A'(t)^T.\left(A(t).A(t)^T\right)^{-1} - A(t)^T.\left(A(t).A(t)^T\right)^{-1}.A(t).A'(t)^T.\left(A(t).A(t)^T\right)^{-1} - A(t)^T.\left(A(t).A(t)^T\right)^{-1}.A'(t).A(t)^T.\left(A(t).A(t)^T\right)^{-1}$$

For example, with

```
In[14]:= A(t_) = {{t, t^3, 2}, {0, -1, E^t}};
```

$B(t) = \left(A(t).A(t)^T\right)^{-1}$ becomes

```
In[15]:= B(t) // Simplify
```

$$\text{Out[15]}= \begin{pmatrix} \frac{1+e^{2t}}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} & \frac{t^3-2e^t}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} \\ \frac{t^3-2e^t}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} & \frac{t^6+t^2+4}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} \end{pmatrix}$$

and the Moore–Penrose pseudoinverse is

```
In[16]:= A⁺(t) // Simplify
```

$$\text{Out[16]}= \begin{pmatrix} \frac{(1+e^{2t})t}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} & \frac{t^4-2e^t t}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} \\ \frac{e^t(e^t t^3+2)}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} & -\frac{2e^t t^3+t^2+4}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} \\ \frac{e^t t^3+2}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} & \frac{t^2(2t+e^t(t^4+1))}{e^{2t}t^6+4e^t t^3+(1+e^{2t})t^2+4} \end{pmatrix}$$

The expression for $\frac{d(A^+(t))}{dt}$ in terms of $B(t)$ and $A^+(t)$ reads as follows.

```
In[17]:= simplified
```

$$\text{Out[17]}= \begin{pmatrix} 1 & 0 \\ 3t^2 & 0 \\ 0 & e^t \end{pmatrix}.B(t) - A^+(t).\begin{pmatrix} 3t^5+t & 2e^t \\ -3t^2 & e^{2t} \end{pmatrix}.B(t) - A^+(t).\begin{pmatrix} 1 & 3t^2 & 0 \\ 0 & 0 & e^t \end{pmatrix}.A^+(t)$$

It is straightforward to check that this result agrees with direct differentiation of $A^+(t)$.

```
In[18]:= Simplify[∂A⁺(t)/∂t == ReleaseHold[simplified]]
```

```
Out[18]= True
```

TROTT'S CORNER

Visualization of Riemann Surfaces IId

Compositions of Elementary Transcendental Functions

Michael Trott

Wolfram Research, Inc.
mtrott@wolfram.com

This is the last installment of my series on Riemann surfaces of compositions of elementary transcendental functions. In the previous installment, *Mathematica* code to calculate and visualize Riemann surfaces was developed. Today we will enjoy the fruits of this work and calculate and display a set of example surfaces.

■ Some Examples

Here is a collection of two dozen Riemann surfaces of "arbitrary" functions calculated using the implemented function **`RiemannSurface`**. Nearly any "arbitrary" function can be drawn, but to avoid the surface intersecting itself too often, it should typically not contain more than three logarithms. The following examples were selected according to the "appearance" of their Riemann surface from a few hundred "randomly" generated functions and their rendered Riemann surfaces. The implementation is quite general and works for arbitrary functions. The author encourages the user to try their own favorite function.

□ #01

This function inherits its 2×3 branch points from the arcsine function and the three roots of the cubic polynomial. The denominator contributes a pole at $z = 1$.

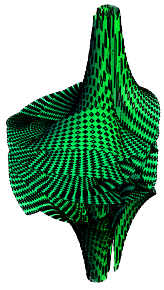
```
In[1]:= wExample = ArcSin[z^3 + 3 z - 7] / (z - 1);
```

In[2]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[2]=
```
{Root[-6 + 3 #1 + #1^3 &, 1], Root[-6 + 3 #1 + #1^3 &, 3],
 Root[-6 + 3 #1 + #1^3 &, 2], Root[-8 + 3 #1 + #1^3 &, 1],
 Root[-8 + 3 #1 + #1^3 &, 3], Root[-8 + 3 #1 + #1^3 &, 2]}
```

In[3]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.38], Hue[0.38], 1.23],
    RiemannSurface[{wExample, z}, Im, {60, 28}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.6, NewPolygonShape → CutVertices]}],
  BoxRatios → {1, 1, 2}, Boxed → False];
```

From In[3]:=



□ #02

This function has five logarithmic branch points. They arise from the argument of the logarithm.

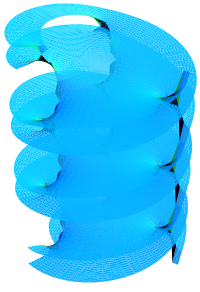
In[4]:=
```
wExample = Log[Sqrt[z^3 - 4] + z];
```

In[5]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[5]=
$$\left\{(-2)^{2/3},\ 2^{2/3},\ \text{Root}[-4+\#1^3\ \&,\ 2],\ -\frac{1}{2}\,i\left(-i+\sqrt{7}\right),\ \frac{1}{2}\,i\left(i+\sqrt{7}\right)\right\}$$

In[6]:=
```
Show[
  Graphics3D[{EdgeForm[], SurfaceColor[Hue[0.45], Hue[0.69], 1.82],
    RiemannSurface[{wExample, z}, Im, {60, 28}, -1, 1]}],
  BoxRatios → {1, 1, 3/2}, Boxed → False];
```

From In[6]:=



□ #03

This function has four branch points and a clearly visible logarithmic singularity at the origin.

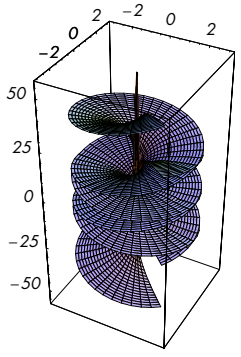
```
In[7]:= wExample = Sqrt[2 Log[z]^3 - 1];

In[8]:= branchPoints[wExample,z] // FullSimplify

Out[8]= {0, e^(-(-1/2)^(1/3)), e^(1/2^(1/3)), e^Root[-1+2#1^3&,3]}

In[9]:= Show[Graphics3D[{EdgeForm[Thickness[0.0001]],
        SurfaceColor[Hue[0.12], Hue[0.64], 1.08],
        RiemannSurface[{wExample, z}, Im, {40, 20}, -1, 1]}],
      Axes → True, BoxRatios → {1, 1, 2}];
```

From In[9]:=



□ #04

This function has six branch points which are easy to see in the resulting graphic.

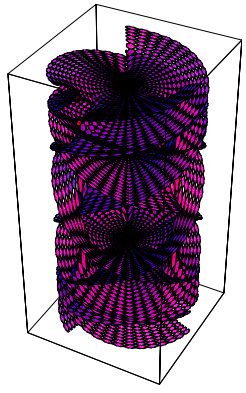
```
In[10]:= wExample = Sqrt[ArcCosh[z^3 - 2]];

In[11]:= branchPoints[wExample,z] // FullSimplify

Out[11]= {1, -(-3)^(1/3), -(-1)^(1/3), (-1)^(2/3), 3^(1/3), Root[-3 + #1^3 &, 3]}
```

In[12]:=
```
Show[Graphics3D[{EdgeForm[Thickness[0.0001]],
    SurfaceColor[Hue[0.8], Hue[0.0], 2.4],
    RiemannSurface[{wExample, z}, Im, {40, 20}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.7, NewPolygonShape → CutVertices]}],
  Axes → False, BoxRatios → {1, 1, 2}];
```

From In[12]:=



□ #05

This function has two branch points at $z = 0$ and $z = 1 + 1/e \approx 1.37$. Both are clearly visible.

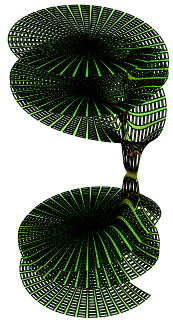
In[13]:=
```
wExample = √(Log[z - 1] + 1);
```

In[14]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[14]=
$$\left\{1,\ 1 + \frac{1}{e}\right\}$$

In[15]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.22], Hue[0.36], 1.54],
    RiemannSurface[{wExample, z}, Re, {40, 20}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.72, NewPolygonShape → OuterConcentric]}],
  BoxRatios → {1, 1, 2}, Boxed → False];
```

From In[15]:=



#06

This function has logarithmic and square root branch points.

In[16]:=
```
wExample = ArcSin[z^2 + Sqrt[z] - 1];
```

In[17]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[17]=
```
{0, 1, -(-1)^(1/3), (-1)^(2/3),
 Root[-4 - 3 #1 + #1^2 + #1^3 &, 3], Root[-4 - 3 #1 + #1^2 + #1^3 &, 2]}
```

In[18]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.44], Hue[0.48], 1.08],
    RiemannSurface[{wExample, z}, Im, {40, 20}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.7, NewPolygonShape → VerticesTriangles]}],
  Boxed → False, BoxRatios → {1, 1, 2}];
```

From In[18]:=



#07

Here is another function with logarithmic and square root branch points.

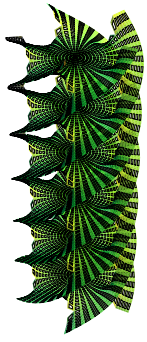
In[19]:=
```
wExample = Log[Sqrt[z^5 - 1]];
```

In[20]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[20]=
```
{1, -(-1)^(1/5), (-1)^(2/5), -(-1)^(3/5), (-1)^(4/5)}
```

In[21]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.37], Hue[0.06], 1.48],
    RiemannSurface[{wExample, z}, Im, {40, 20}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.5, NewPolygonShape → OuterConcentric]}],
  Boxed → False, BoxRatios → {1, 1, 5/2}];
```

From In[21]:=



□ #08

This function has five branch points. The three visible singularities arise from the zeros of the cubic polynomial in the denominator.

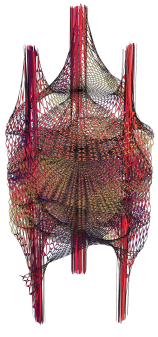
In[22]:=
$$\text{wExample} = \frac{\text{Log}[z^2 - I]}{\sqrt[3]{z^3 - 1}};$$

In[23]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[23]=
$$\left\{1, -(-1)^{1/4}, (-1)^{1/4}, -(-1)^{1/3}, (-1)^{2/3}\right\}$$

In[24]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.91], Hue[0.16], 1.96],
    RiemannSurface[{wExample, z}, Im, {40, 20}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.7, NewPolygonShape → VerticesTriangles]}],
  Boxed → False, BoxRatios → {1, 1, 2.2}];
```

From In[24]:=



#09

This is a function with logarithmic branch points and a logarithmic singularity.

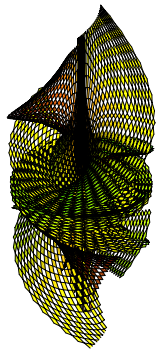
```
In[25]:= wExample = (Log[z^2]^7 - 1)^(1/3);

In[26]:= branchPoints[wExample,z] // FullSimplify

Out[26]= {0, -Sqrt[e], Sqrt[e], -e^(-1/2 (-1)^(1/7)), e^(-1/2 (-1)^(1/7)), -e^(1/2 (-1)^(2/7)),
    e^(1/2 (-1)^(2/7)), -e^(-1/2 (-1)^(3/7)), e^(-1/2 (-1)^(3/7)), -e^(1/2 (-1)^(4/7)),
    e^(1/2 (-1)^(4/7)), -e^(-1/2 (-1)^(5/7)), e^(-1/2 (-1)^(5/7)), -e^(1/2 (-1)^(6/7)),
    e^(1/2 (-1)^(6/7))}

In[27]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
      SurfaceColor[Hue[0.22], Hue[0.15], 1.77],
      RiemannSurface[{wExample, z}, Im, {36, 20}, -1, 1] /.
       p_Polygon :> holePolygon[p, 1, NewPolygonShape -> CutVertices]}],
    Boxed -> False, BoxRatios -> {1, 1, 2.2}, PlotRange -> {-200, 200}];
```

From In[27]:=



#10

This is a function with six branch points.

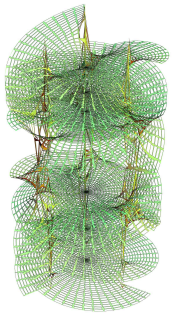
```
In[28]:= wExample = (Log[z^3 - 1] - 1)^(1/3);

In[29]:= branchPoints[wExample,z] // FullSimplify

Out[29]= {1, -(-2)^(1/3), -(-1)^(1/3), (-1)^(2/3), 2^(1/3), Root[-2 + #1^3 &, 3]}
```

In[30]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.16], Hue[0.44], 1.32],
    RiemannSurface[{wExample, z}, Im, {46, 26}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.7, NewPolygonShape → RandomQuadrilateral]}],
  Boxed → False, BoxRatios → {1, 1, 2}];
```

From In[30]:=



□ #11

The two branch points of the following function are $z = \pm 1$.

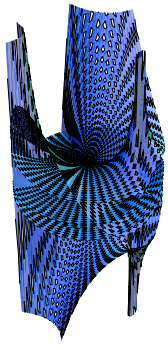
In[31]:=
```
wExample = (√(z - 1))^√(z+1) - 1;
```

In[32]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[32]=
```
{-1, 1}
```

In[33]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.52], Hue[0.76], 1.62],
    RiemannSurface[{wExample, z}, Im, {50, 26}, -1, 1] /. p_Polygon :>
      holePolygon[p, 0.6, NewPolygonShape → CutVertices]}],
  Boxed → False, BoxRatios → {1, 1, 2}, Axes → False,
  PlotRange -> {-7, 7}];
```

From In[33]:=



□ #12

This is a more complicated function with two logarithmic branch points.

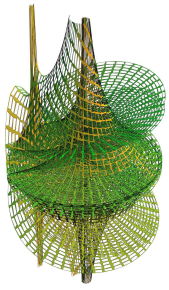
```
In[34]:= wExample = Log[z - (z + 1)^(1/3) + 1]^2;

In[35]:= branchPoints[wExample,z] // FullSimplify

Out[35]= {-1, 0}

In[36]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
      SurfaceColor[Hue[0.34], Hue[0.98], 1.59],
      RiemannSurface[{wExample, z}, Im, {46, 26}, -1, 1] /. p_Polygon :>
        holePolygon[p, 0.6, NewPolygonShape → RandomQuadrilateral]}],
    Boxed → False, BoxRatios → {1, 1, 2}];
```

From In[36]:=



□ #13

This is a complicated function involving nested square roots and a logarithm.

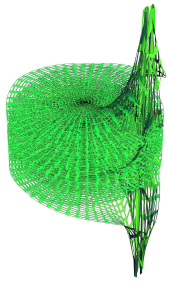
```
In[37]:= wExample = Sqrt[(Log[z + 1] - 1)/(Sqrt[z - 1] + 1)];

In[38]:= branchPoints[wExample,z] // FullSimplify

Out[38]= {-1, 1, -1 + e}

In[39]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
      SurfaceColor[Hue[0.39], Hue[0.26], 1.59],
      RiemannSurface[{wExample, z}, Re, {48, 24}, -1, 1] /. p_Polygon :>
        holePolygon[p, 0.5, NewPolygonShape → RandomQuadrilateral]}],
    Boxed → False, BoxRatios → {1, 1, 2}, PlotRange → {-8, 8}];
```

From In[39]:=



□ #14

This is a function with five logarithmic branch points.

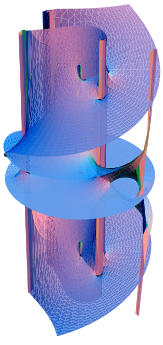
In[40]:=
$$\text{wExample} = \text{Log}\left[\frac{z - \frac{1}{2}}{\sqrt{z} + z^2 - 1}\right]^3 ;$$

In[41]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[41]=
$$\left\{0, \frac{1}{2}, \text{Root}[1 - \#1 - 2\#1^2 + \#1^4 \&, 3],\right.$$
$$\left.\text{Root}[1 - \#1 - 2\#1^2 + \#1^4 \&, 4], \text{Root}[1 - \#1 - 2\#1^2 + \#1^4 \&, 1]\right\}$$

In[42]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.55], Hue[0.02], 1.32],
    RiemannSurface[{wExample, z}, Im, {52, 24}, -1, 1]}],
  Boxed → False, BoxRatios → {1, 1, 2}, PlotRange → {-350, 350}];
```

From In[42]:=



□ #15

Here is an algebraic function with seven branch points.

In[43]:=
$$\text{wExample} = \sqrt[5]{z^7 - 2z^4 - 3z + 8} ;$$

In[44]:=
```
branchPoints[wExample,z] // FullSimplify
```

Out[44]=
$$\{\text{Root}[8 - 3\#1 - 2\#1^4 + \#1^7 \&, 1],$$
$$\text{Root}[8 - 3\#1 - 2\#1^4 + \#1^7 \&, 2], \text{Root}[8 - 3\#1 - 2\#1^4 + \#1^7 \&, 3],$$
$$\text{Root}[8 - 3\#1 - 2\#1^4 + \#1^7 \&, 4], \text{Root}[8 - 3\#1 - 2\#1^4 + \#1^7 \&, 5],$$
$$\text{Root}[8 - 3\#1 - 2\#1^4 + \#1^7 \&, 6], \text{Root}[8 - 3\#1 - 2\#1^4 + \#1^7 \&, 7]\}$$

```
In[45]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.85], Hue[0.21], 2.42],
           RiemannSurface[{wExample, z}, Im, {48, 24}, -1, 1]}],
         Boxed → False, BoxRatios → {1, 1, 2}, PlotRange → All];
```

From In[45]:=



□ #16

Here is another logarithm with a complicated algebraic argument.

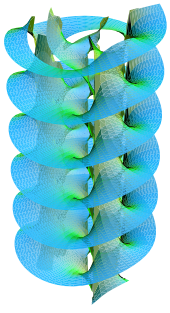
```
In[46]:= wExample = Log[(z^3 - 2 z + 7)/Sqrt[z^5 - 3 z^3 + z - 1]];
```

```
In[47]:= branchPoints[wExample,z] // FullSimplify
```

```
Out[47]= {-1, Root[7 - 2 #1 + #1^3 &, 1], Root[7 - 2 #1 + #1^3 &, 3],
          Root[7 - 2 #1 + #1^3 &, 2], Root[-1 + 2 #1 - 2 #1^2 - #1^3 + #1^4 &, 3],
          Root[-1 + 2 #1 - 2 #1^2 - #1^3 + #1^4 &, 4],
          Root[-1 + 2 #1 - 2 #1^2 - #1^3 + #1^4 &, 1],
          Root[-1 + 2 #1 - 2 #1^2 - #1^3 + #1^4 &, 2]}
```

```
In[48]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.32], Hue[0.76], 1.91],
           RiemannSurface[{wExample, z}, Im, {52, 26}, -1, 1]}],
         Boxed → False, BoxRatios → {1, 1, 2}, PlotRange → {-10, 10}];
```

From In[48]:=



#17

This is a rational function containing an inverse trigonometric function.

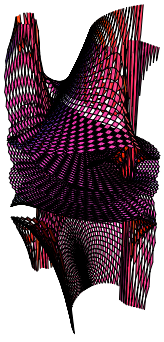
```
In[49]:= wExample = 1 / (ArcSin[z^2 - 5] + z);

In[50]:= branchPoints[wExample,z] // FullSimplify

Out[50]= {-2, 2, -Sqrt[6], Sqrt[6]}

In[51]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.07], Hue[0.85], 0.8],
           RiemannSurface[{wExample, z}, Im, {48, 24}, -1, 1]}] /.
         p_Polygon :> holePolygon[p, 1, NewPolygonShape -> CutVertices],
       Boxed -> False, BoxRatios -> {1, 1, 2},
       PlotRange -> {-1 / 3, 1 / 3}];
```

From In[51]:=



#18

This function has coinciding logarithmic and square root branch points at $z = 0$.

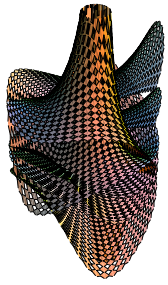
```
In[52]:= wExample = (1 + z + z^2 + Sqrt[z]) / (1 + z + z^2 + Log[z]);

In[53]:= branchPoints[wExample,z] // FullSimplify

Out[53]= {0}
```

```
In[54]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.22], Hue[0.81], 1.95],
           RiemannSurface[{wExample, z}, Im, {52, 26}, -1, 1]}] /.
         p_Polygon :> holePolygon[p, 0.8, NewPolygonShape -> CutVertices],
        Boxed -> False, BoxRatios -> {1, 1, 2}, PlotRange -> {-1, 1}];
```

From In[54]:=



□ #19

This quite complicated function has five branch points.

```
In[55]:= wExample = Cos[Sqrt[z] + (Sqrt[z^2, 3] - z^2)/Log[z^4 - 2 z + 8]];
```

```
In[56]:= branchPoints[wExample,z] // FullSimplify
```

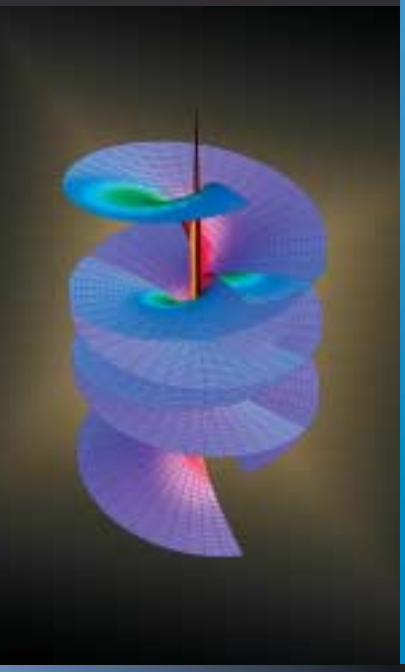
```
Out[56]= {0, Root[8 - 2 #1 + #1^4 &, 1], Root[8 - 2 #1 + #1^4 &, 2],
          Root[8 - 2 #1 + #1^4 &, 3], Root[8 - 2 #1 + #1^4 &, 4]}
```

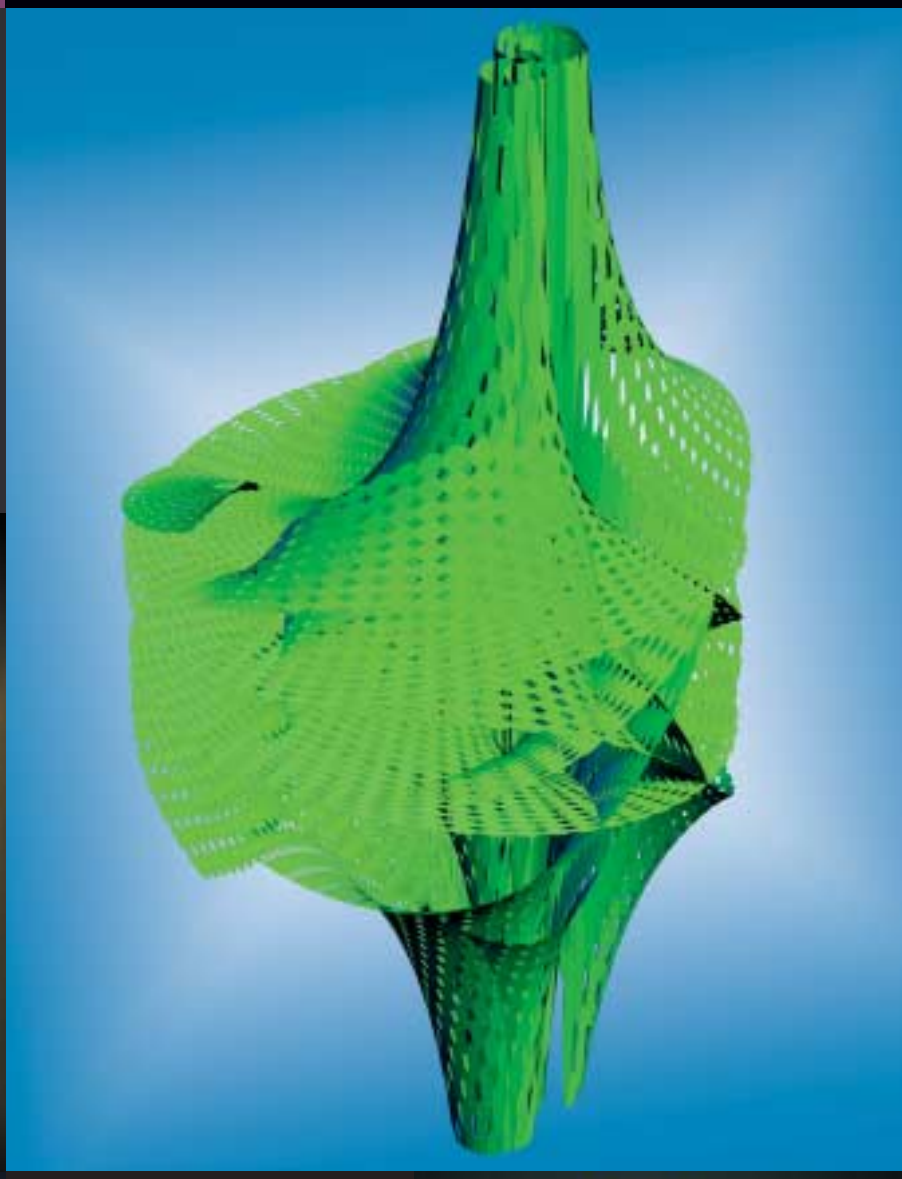
```
In[57]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.28], Hue[0.82], 1.282],
           RiemannSurface[{wExample, z}, Im, {48, 24}, -1, 1]}] /.
         p_Polygon :> holePolygon[p, 1, NewPolygonShape -> CutVertices],
        Boxed -> False, BoxRatios -> {1, 1, 2}, PlotRange -> Automatic];
```

From In[57]:=











Clockwise from the upper left:

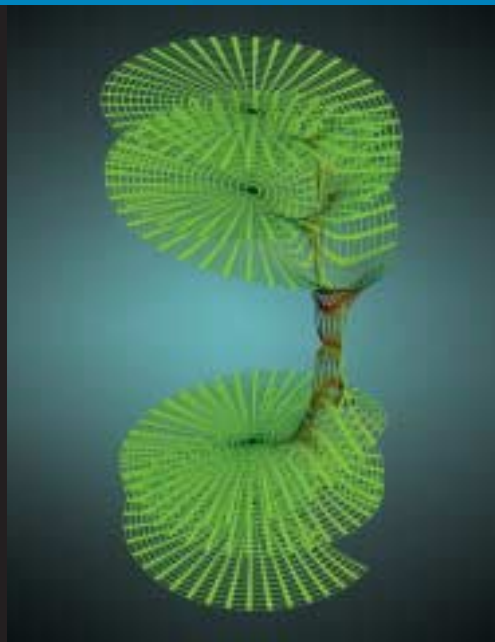
1) $w(z) = \ln\left(z + \sqrt{z^3 - 4}\right)$

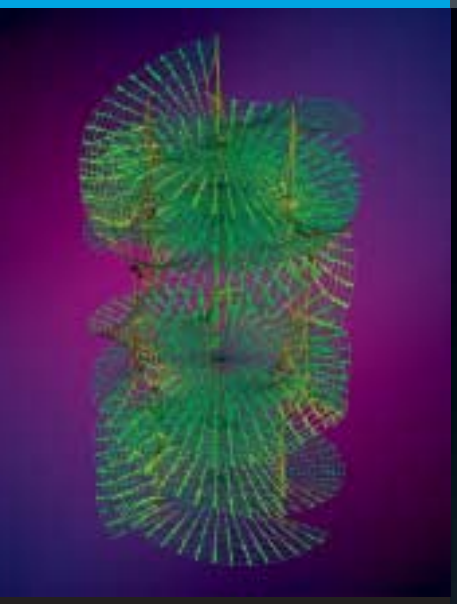
2) $w(z) = \dfrac{\arcsin(z^3 + 3z - 7)}{z - 1}$

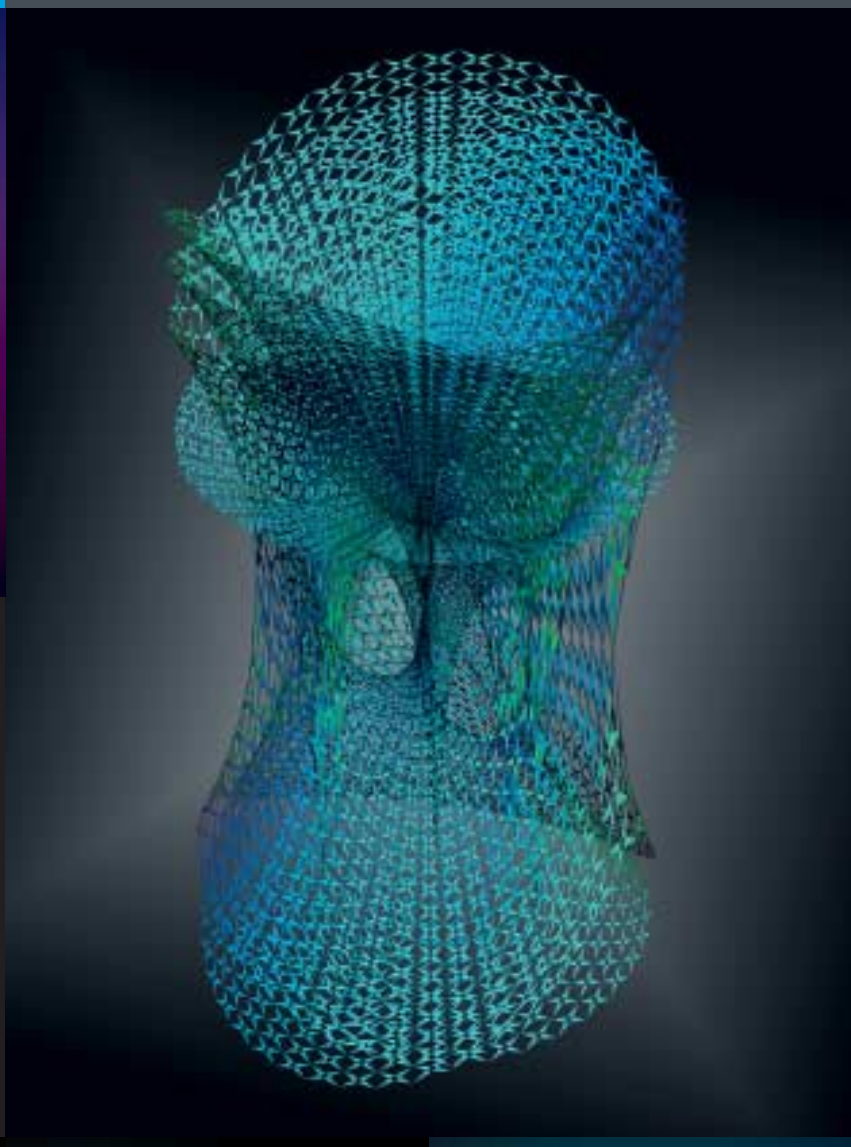
3) $w(z) = \sqrt{\ln(z - 1) + 1}$

4) $w(z) = \sqrt{\operatorname{arccosh}(z^3 - 2)}$

5) $w(z) = \sqrt{2\ln^3(z) - 1}$







Clockwise from the upper left:

1) $w(z) = \sqrt[3]{\ln(z^3 - 1)} - 1$

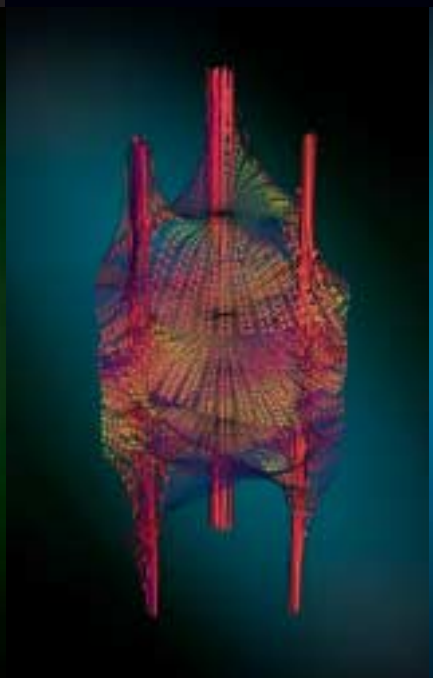
2) $w(z) = \arcsin\left(z^2 + \sqrt{z} - 1\right)$

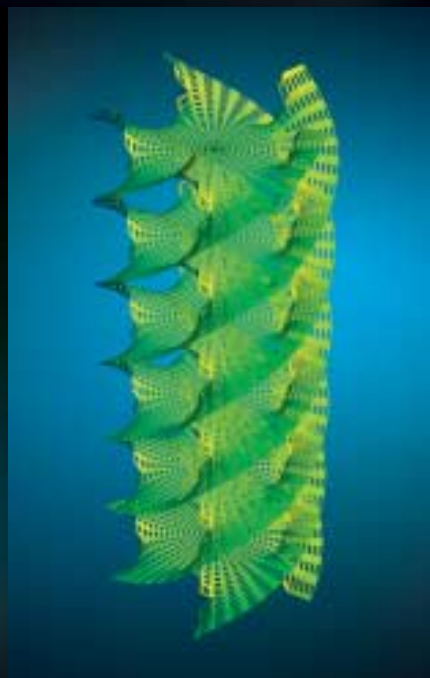
3) $w(z) = \ln\left(\sqrt{z^5 - 1}\right)$

4) $w(z) = \dfrac{\ln(z^2 - i)}{\sqrt[3]{z^3 - 1}}$

5) $w(z) = \sqrt[3]{\ln^7(z^2) - 1}$









Clockwise from the upper left:

1) $w(z) = \sqrt{z-1}^{\sqrt{z+1}} - 1$

2) $w(z) = \ln^2\left(-\sqrt[3]{z+1} + z + 1\right)$

3) $w(z) = \sqrt{\dfrac{\ln(z+1)-1}{\sqrt{z-1}+1}}$

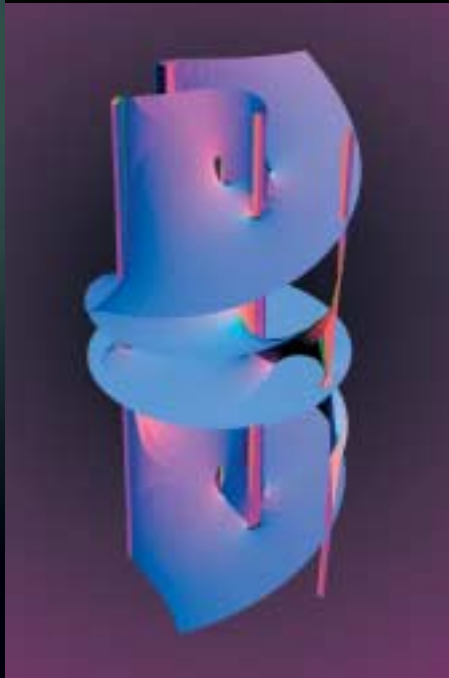
4) $w(z) = \ln^3\left(\dfrac{z-\frac{1}{2}}{z^2+\sqrt{z}-1}\right)$

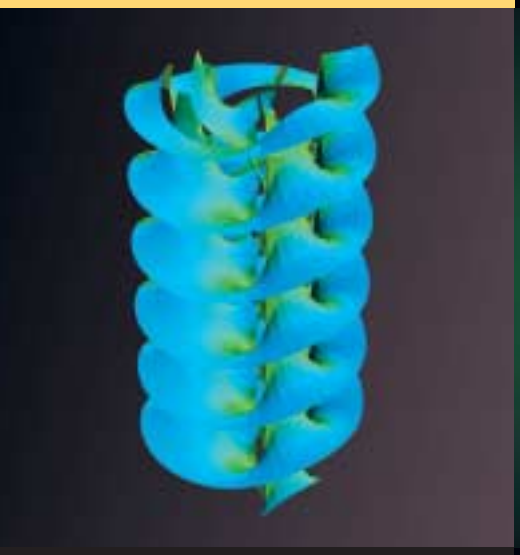
5) $w(z) = \sqrt[5]{z^7 - 2z^4 - 3z + 8}$



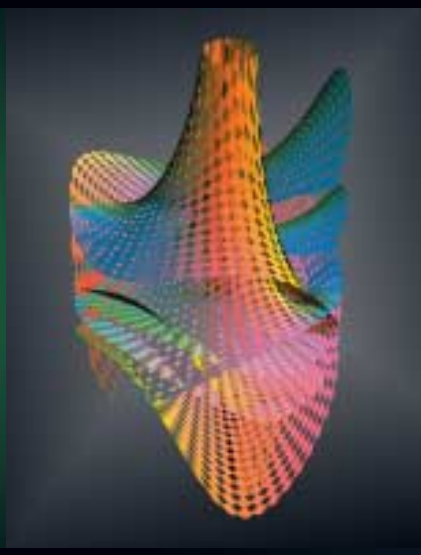












Clockwise from the upper left:

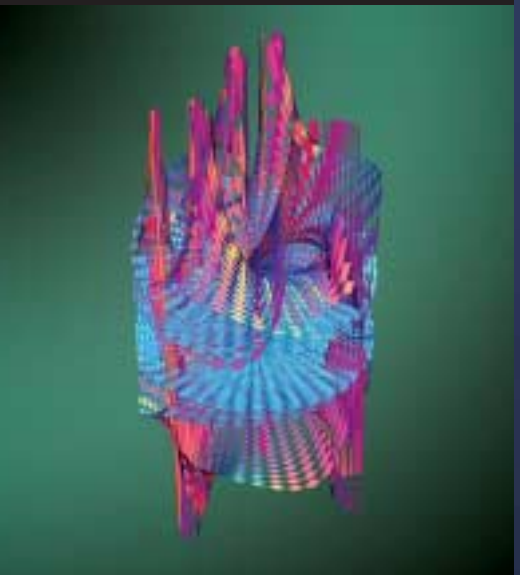
1) $w(z) = \ln\left(\dfrac{z^3 - 2z + 7}{\sqrt{z^5 - 3z^3 + z - 1}}\right)$

2) $w(z) = \dfrac{1}{z + \arcsin(z^2 - 5)}$

3) $w(z) = \dfrac{z^2 + z + \sqrt{z} + 1}{z^2 + z + \ln(z) + 1}$

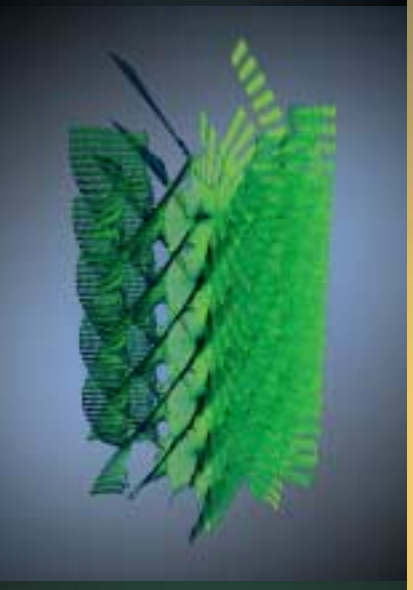
4) $w(z) = \cos\left(\dfrac{\sqrt[3]{z^2} - z^2}{\ln(z^4 - 2z + 8)} + \sqrt{z}\right)$

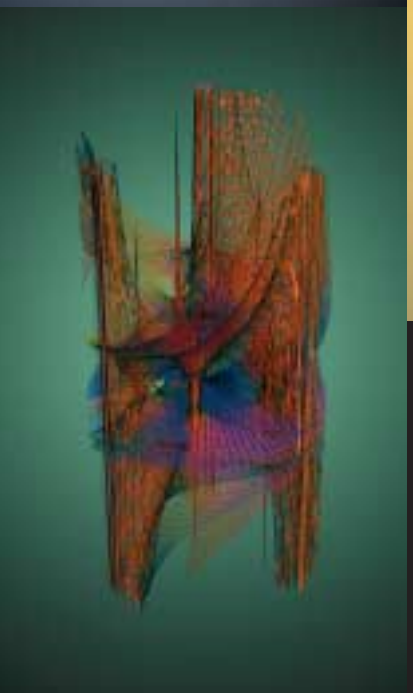
5) $w(z) = \dfrac{1}{\ln\left(z + \frac{1}{\sqrt{z}}\right) + \sqrt{z^2 - 1} + 1}$

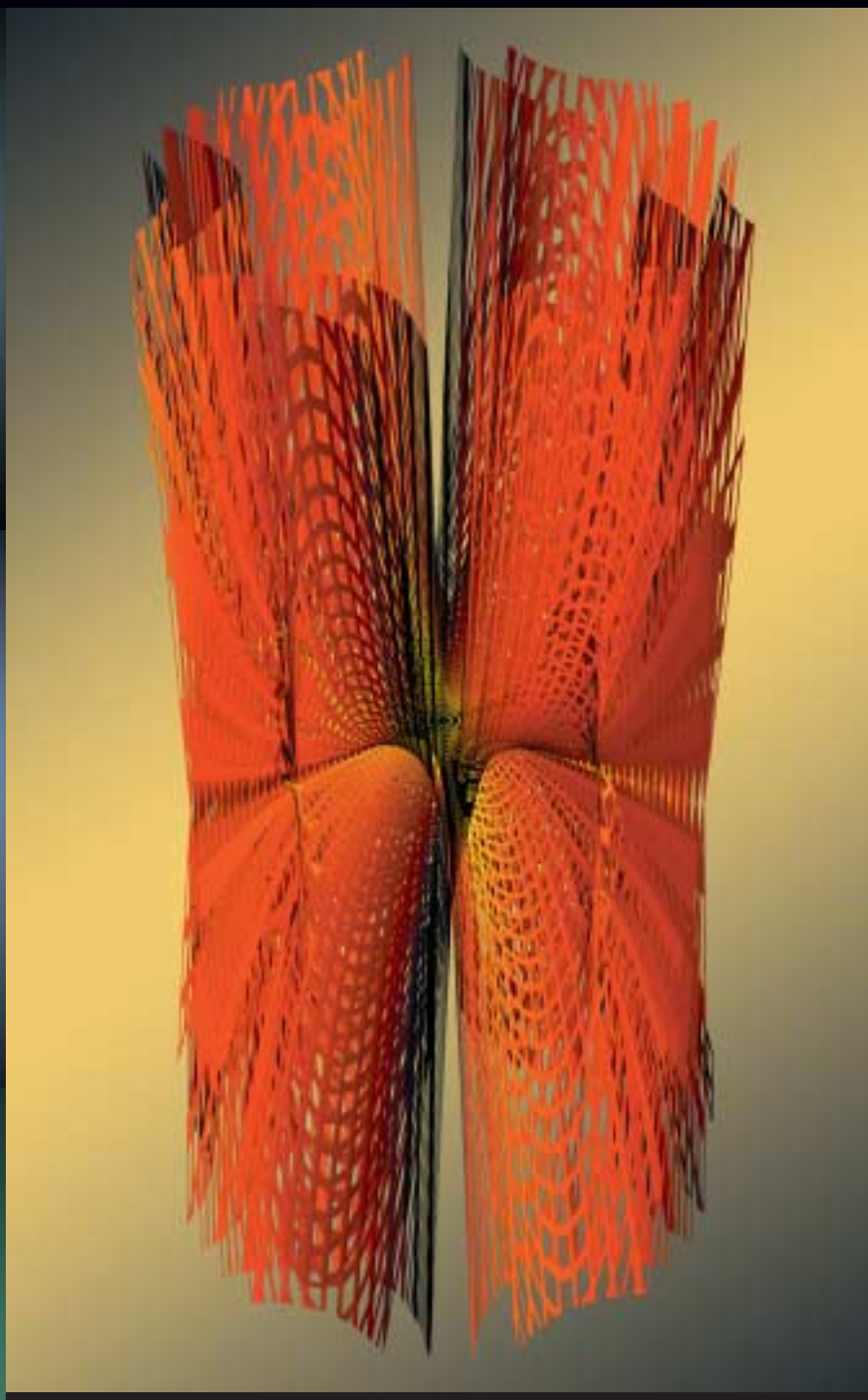












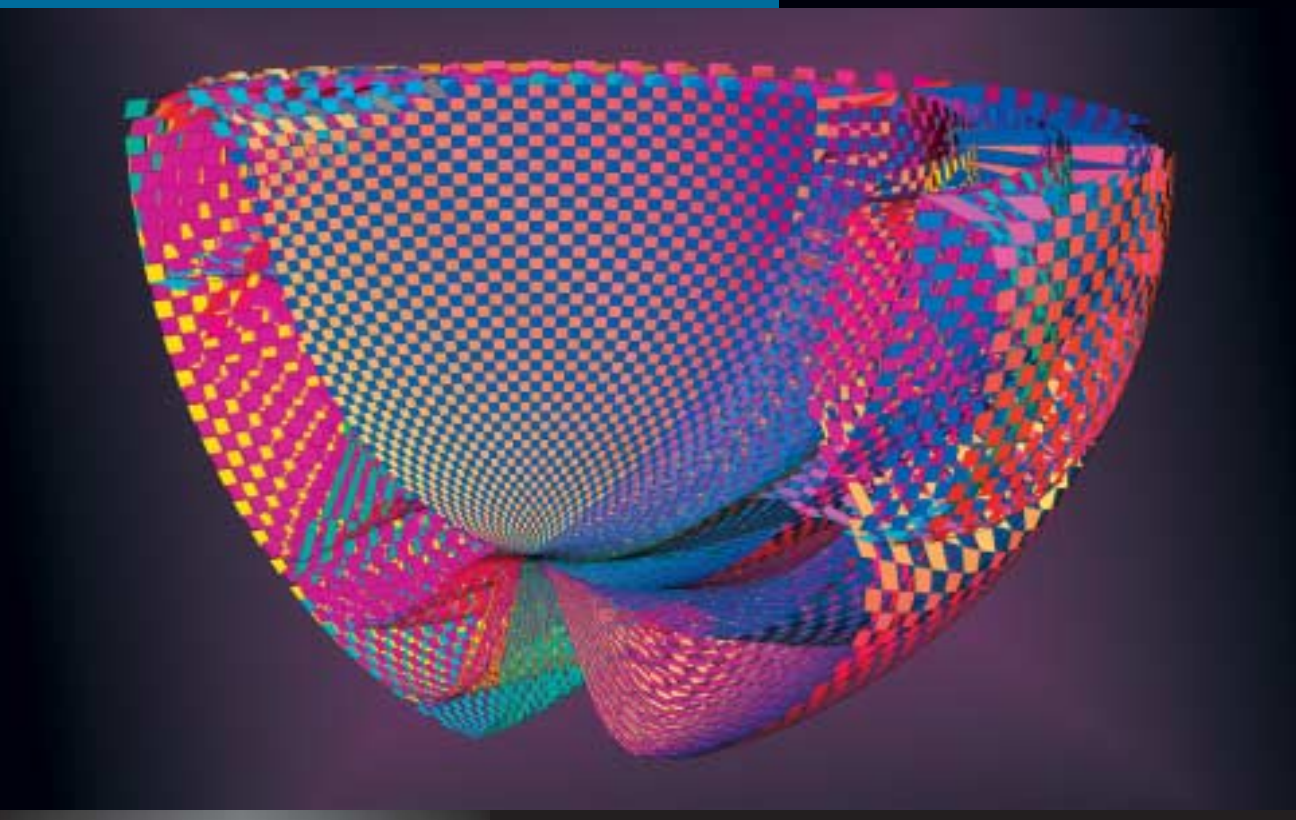
Clockwise from the upper left:

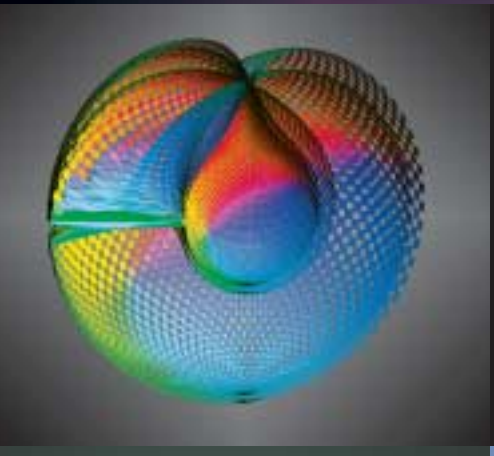
1) $w(z) = \dfrac{\log\left(\frac{z^3+3z+7}{z^4-4z^3+7z^2-8}\right)}{z^3-2z+5}$

2) $w(z) = \dfrac{z^3-1}{\arctan(z^3-1)}$

3) $w(z) = \dfrac{\ln(z^5-1)+2}{\ln(z^3-2)-1}$

4) $w(z) = \ln(z^{17}-12z^{11}+3z+4)$





Clockwise from upper left:

1) $w(z) = z^{z^z}$

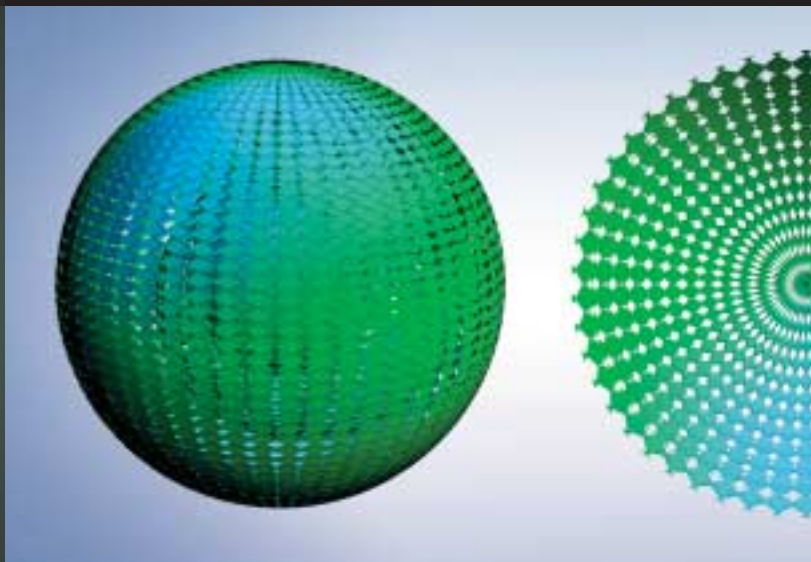
2) $w(z) = \sqrt{\ln(z)}$

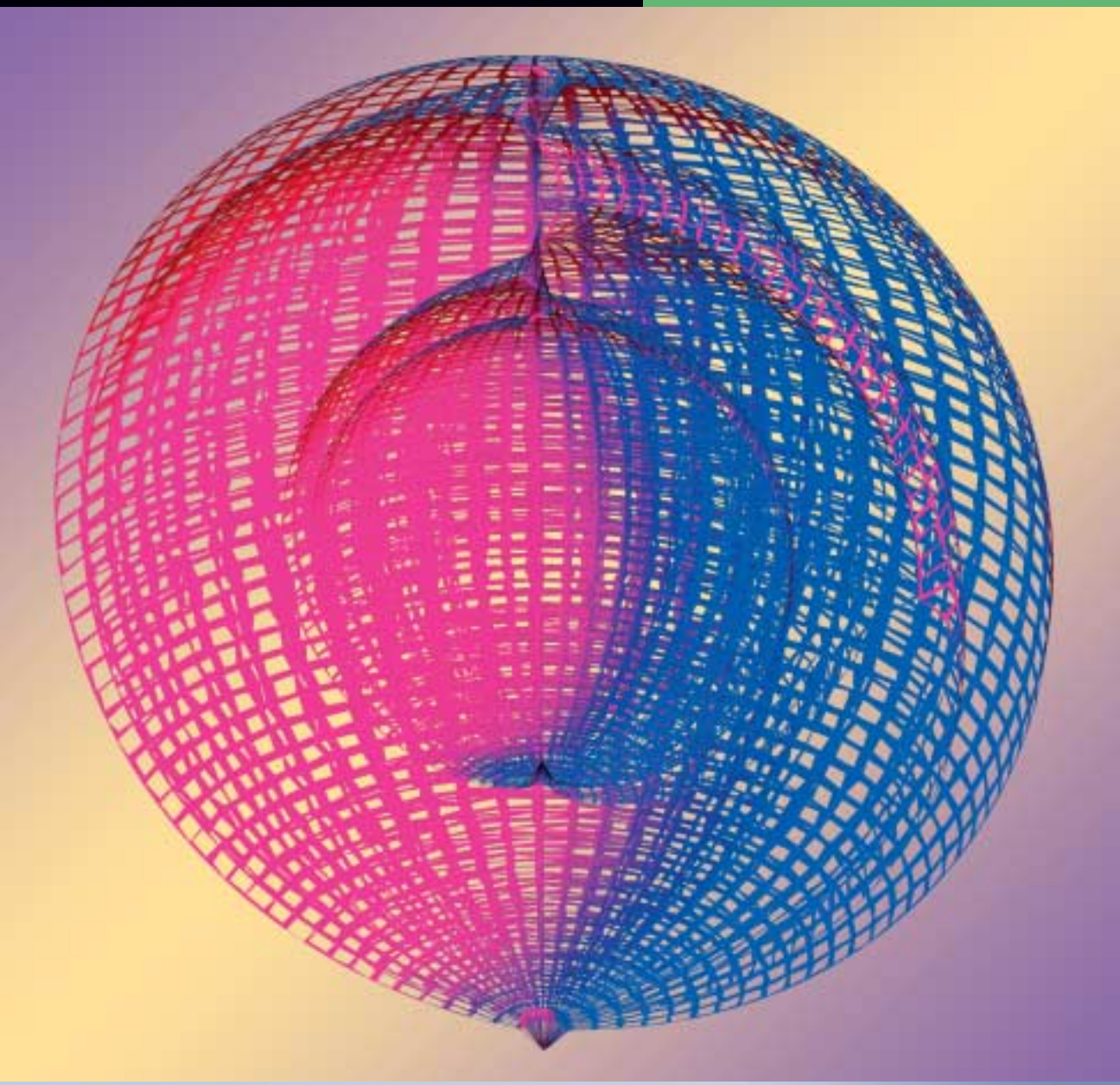
3) Four views of
$w(z) = \arcsin(z)$

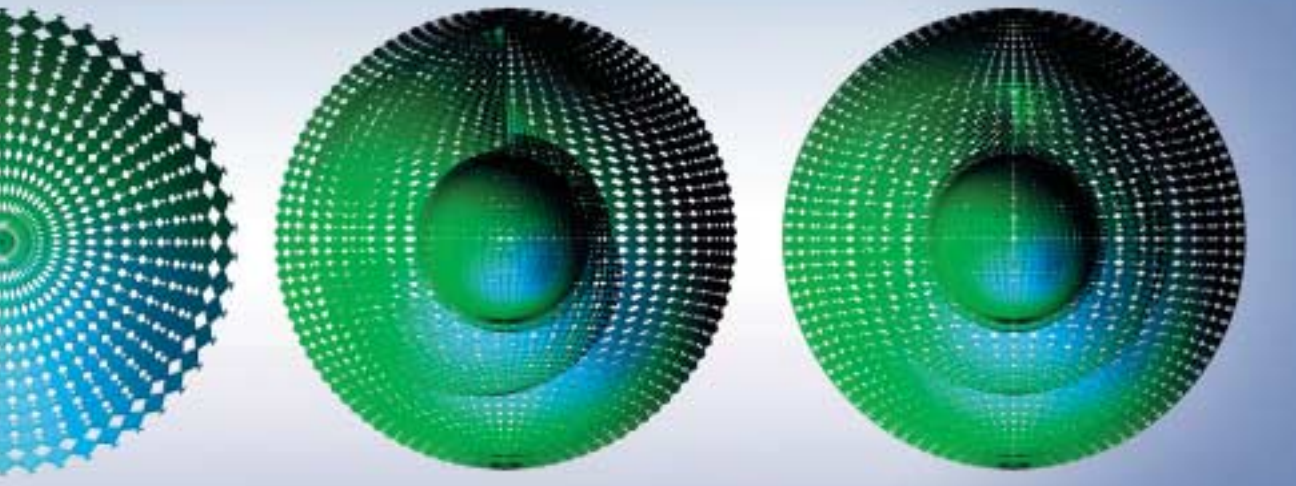
4) $w(z) = \ln(z)$

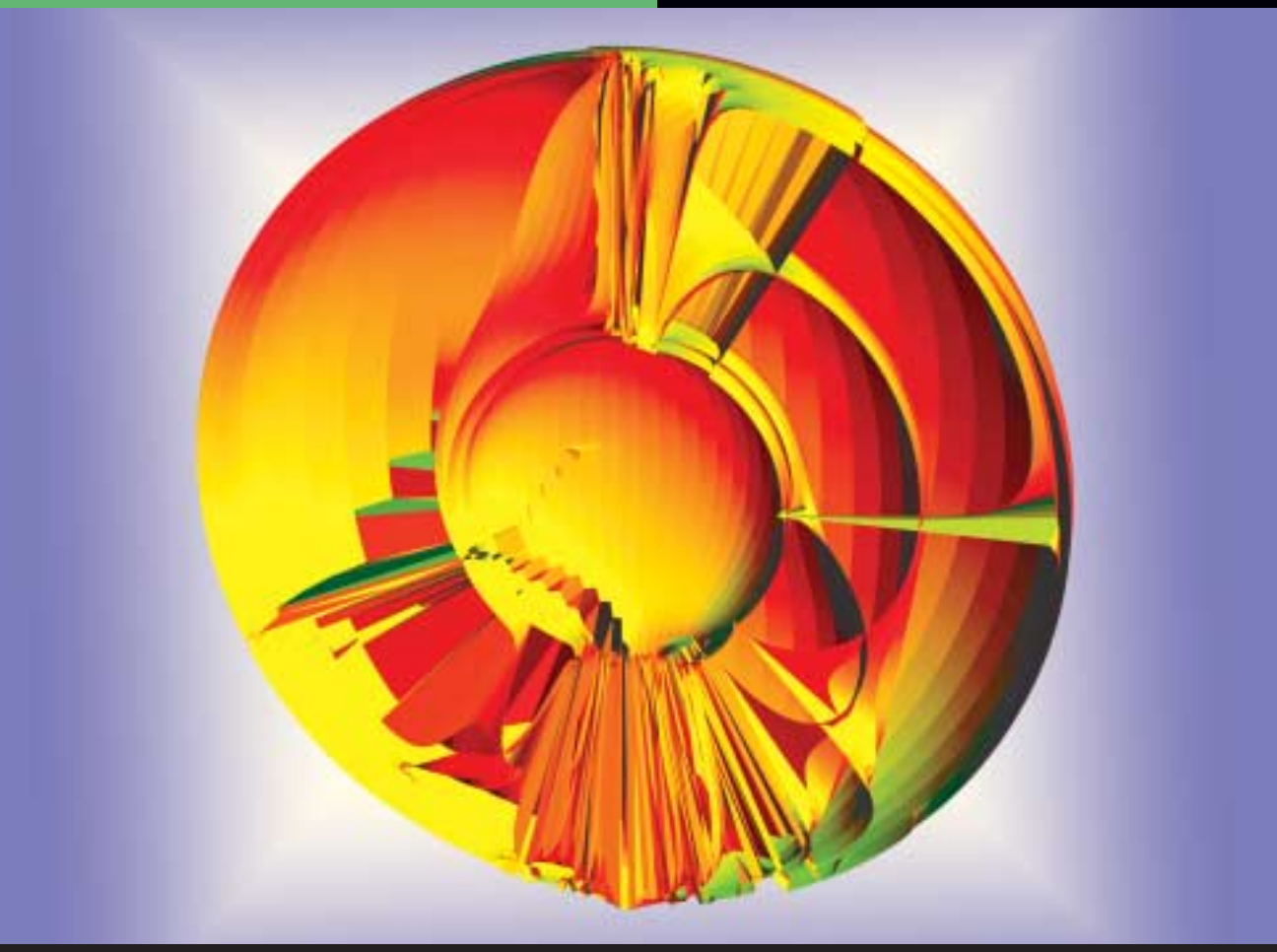
5) $w(z) = \dfrac{\ln(z+1)}{-z-1} + \dfrac{\ln(-z-1)}{z+1}$









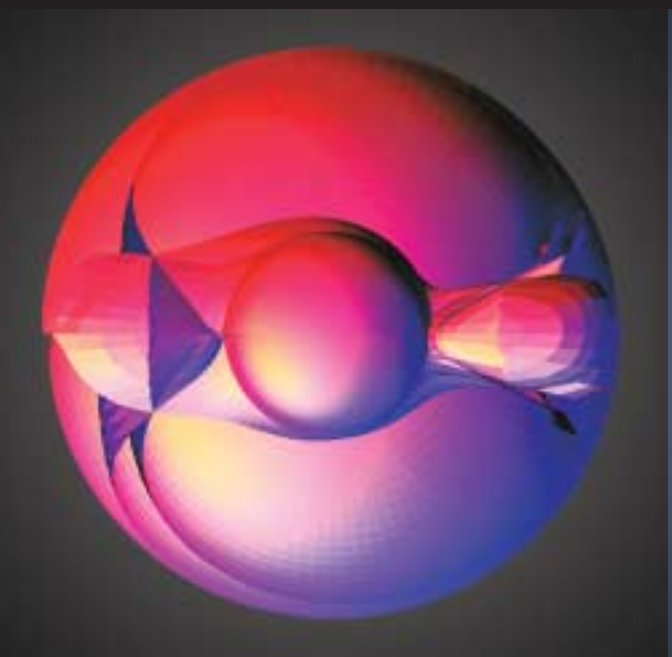


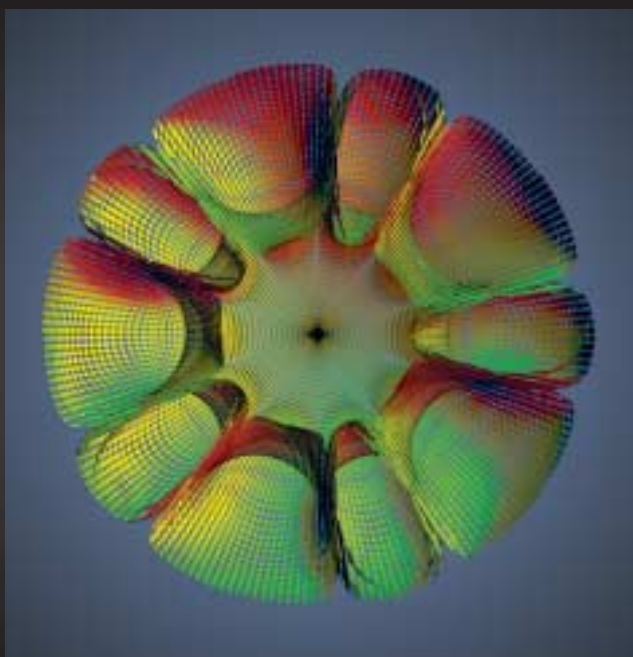
Clockwise from above:

1) $w(z) = \ln\left(\sin\left(\sqrt{z} + \frac{1}{z}\right)\right)$

2) $w(z) = \ln^2(z^5 - 1)$

3) $w(z) = \ln^2\left(z - 1 + \sqrt[3]{z+1}\right)$





□ #20

The five branch points of this function are easily identifiable in the graphic.

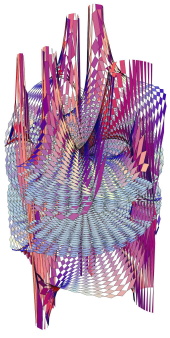
```
In[58]:= wExample = 1 / (1 + Log[z + 1/Sqrt[z]] + Sqrt[z^2 - 1]);

In[59]:= branchPoints[wExample,z] // FullSimplify

Out[59]= {-1, 0, 1, -(-1)^(1/3), (-1)^(2/3)}

In[60]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.84], Hue[0.31], 1.03],
           RiemannSurface[{wExample, z}, Im, {40, 24}, -1, 1]}] /.
         p_Polygon :> holePolygon[p, 1, NewPolygonShape -> CutVertices],
        Boxed -> False, BoxRatios -> {1, 1, 2}, PlotRange -> {-1, 1}];
```

From In[60]:=



□ #21

The logarithm here has a complicated structure that gives rise to seven branch points. This time we use five sheets for each logarithm.

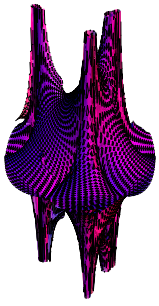
```
In[61]:= wExample = Log[(z^3 + 3 z - 7)/(z^4 - 4 z^3 + 7 z^2 - 8)] / (z^3 - 2 z + 5);

In[62]:= branchPoints[wExample,z] // FullSimplify

Out[62]= {Root[-7 + 3 #1 + #1^3 &, 1],
          Root[-7 + 3 #1 + #1^3 &, 2], Root[-7 + 3 #1 + #1^3 &, 3],
          Root[-8 + 7 #1^2 - 4 #1^3 + #1^4 &, 3], Root[-8 + 7 #1^2 - 4 #1^3 + #1^4 &, 4],
          Root[-8 + 7 #1^2 - 4 #1^3 + #1^4 &, 1], Root[-8 + 7 #1^2 - 4 #1^3 + #1^4 &, 2]}
```

```
In[63]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.78], Hue[0.78], 2.24],
           RiemannSurface[{wExample, z}, Im, {80, 36}, -2, 2] /. p_Polygon :>
             holePolygon[p, 0.6, NewPolygonShape → CutVertices]}],
         BoxRatios → {1, 1, 2}, Boxed → False];
```

From In[63]:=



□ #22

This function has six branch points.

```
In[64]:= wExample = (z^3 - 1)/(ArcTan[z^3 - 1]);

In[65]:= branchPoints[wExample,z] // FullSimplify

Out[65]= {(1 - i)^(1/3), -(-1 + i)^(1/3), (-1)^(2/3) (1 - i)^(1/3),
          (1 + i)^(1/3), -(-1)^(1/3) (1 + i)^(1/3), (-1)^(2/3) (1 + i)^(1/3)}

In[66]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
           SurfaceColor[Hue[0.02], Hue[0.22], 2.84],
           RiemannSurface[{wExample, z}, Im, {80, 36}, -1, 1] /. p_Polygon :>
             holePolygon[p, 0.7, NewPolygonShape → OuterConcentric]}],
         BoxRatios → {1, 1, 2}, Boxed → False];
```

From In[66]:=



□ #23

Here we have eight branch points originating form the zeros of the two polynomials inside the logarithms.

```
In[67]:= wExample = (Log[z^5 - 1] + 2)/(Log[z^3 - 2] - 1);

In[68]:= branchPoints[wExample,z] // FullSimplify

Out[68]= {1, -(-2)^(1/3), -(-1)^(1/5), (-1)^(2/5),
  -(-1)^(3/5), (-1)^(4/5), 2^(1/3), Root[-2 + #1^3 &, 3]}

In[69]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
      SurfaceColor[Hue[0.16], Hue[0.84], 3.04],
      RiemannSurface[{wExample, z}, Im, {80, 36}, -1, 1] /. p_Polygon :>
        holePolygon[p, 0.76, NewPolygonShape -> VerticesTriangles]}],
    BoxRatios -> {1, 1, 2}, Boxed -> False];
```

From In[69]:=



□ #24

The logarithm here has a high-degree polynomial as its argument which gives rise to 17 branch points.

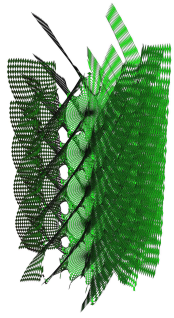
```
In[70]:= wExample = Log[z^17 - 12 z^11 + 3 z + 4];

In[71]:= branchPoints[wExample,z] // FullSimplify

Out[71]= {Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 1],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 2], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 3],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 4], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 5],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 6], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 7],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 8], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 9],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 10], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 11],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 12], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 13],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 14], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 15],
   Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 16], Root[4 + 3 #1 - 12 #1^11 + #1^17 &, 17]}
```

In[72]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
    SurfaceColor[Hue[0.32], Hue[0.36], 2.64],
    RiemannSurface[{wExample, z}, Im, {80, 36}, -2, 2] /.
     p_Polygon :> holePolygon[p, 1, NewPolygonShape → CutVertices]}],
  BoxRatios → {1, 1, 2}, Boxed → False];
```

From In[72]:=



Although many more compositions of transcendental functions can be visualized using the function **RiemannSurface**, we will end here.

■ Riemann Surfaces over the Riemann Sphere

We will construct pictures that show $\mathrm{Im}(w)$ or $\mathrm{Re}(w)$ not over the complex z-plane, but rather over the Riemann sphere. Using the sphere allows us to cover all the z-values more "equally" and to make the branch points at ∞ more visible. Given the Riemann sphere of radius $R = 1/2$ with center $\{0, 0, 1/2\}$, we visualize $f(w(x + i\,y))$ as a point with the same direction as the image of $x + i\,y$ on the Riemann sphere and distance from the origin $r = R + r_{\max} \arctan(f(w(x + i\,y)))$. The arctan in the last formula uniquely maps the interval $\{-\infty, \infty\}$ to a finite interval. The function **RiemannSpherePoint** calculates the position of the point corresponding to the spherical coordinates $\{\varphi, \theta\}$ under the mapping of the function f onto the Riemann sphere, resulting in a "multi-skinned-Riemann egg."

In[73]:=
```
ExtendedRiemannSpherePoint[{φ_, θ_}, f_, rMax_: 1/4] :=
    Module[{x, y, d},
           {x, y} = N[Cot[θ/2] {Cos[φ], Sin[φ]}];
              d = {Cos[φ] Sin[θ], Sin[φ] Sin[θ], Cos[θ]};
              {0, 0, 1/2} + d (1/2 + 2/π rMax ArcTan[f[x, y]])]
```

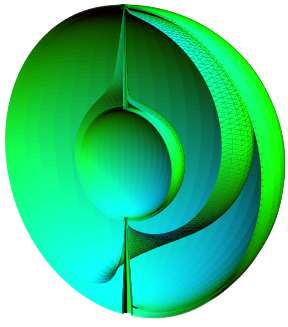
The function **RiemannSpherePatch** generates a patch for the function f in the specified part of the domain of the spherical coordinates.

```
In[74]:= ExtendedRiemannSpherePatch[f_,
           {φ1_, φ2_}, {θ1_, θ2_}, {ppφ_, ppθ_}, rMax_: 1 / 4] :=
         Module[{ε = 10^-12},
           points = Table[ExtendedRiemannSpherePoint[{φ, θ}, f, rMax],
                      {φ, φ1 + ε, φ2 - ε, (φ2 - φ1 - 2 ε) / ppφ},
                {θ, θ1 + ε, θ2 - ε, (θ2 - θ1 - 2 ε) / ppθ}];
            Function[s, Apply[Polygon[Join[#1, Reverse[#2]]] &,
              Transpose /@ Partition[Partition[#, 2, 1] & /@ s, 2, 1],
             {2}]] @ points]
```

We will give some simple examples (in the sense that the functions have a branch cut structure that is easy to determine and to describe) of functions mapped onto the Riemann sphere. Let us start with the simplest function $\ln(z)$ by using the functions defined above. The two branch points at 0 and ∞ are mapped into the points {0, 0, 1} and {0, 0, 0} on the Riemann sphere. The north pole corresponds to complex infinity. We see the symmetry between the north pole and the south pole, which is a reflection of the formula $\log(1/z) = -\log(z)$ (in the last formula, corrections to the branch cut have been ignored).

```
In[75]:= Show[Graphics3D[{EdgeForm[], GrayLevel[0],
           SurfaceColor[Hue[0.32], Hue[0.58], 1.92],
             Table[ExtendedRiemannSpherePatch[
               Function[{x, y}, Im[#] + Re[#] &[Log[x + i y] + 2 k π i]],
               {0, π}, {0, π}, {24, 80}], {k, -3, 3}]}],
         Boxed → False, ViewPoint → {2, -3, 0.8}];
```

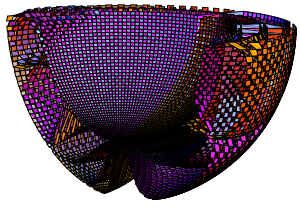
From In[75]:=



The following picture shows some sheets of the Riemann surface of $w(z) = z^{z^z}$ inside the upper half of the unit disk. For easier viewing we color the sheets differently and do not show all the polygons.

```
In[76]:= Show[Graphics3D[{EdgeForm[{GrayLevel[0], Thickness[0.0001]}],
           Table[
         {SurfaceColor[Hue[Random[]], Hue[Random[]], 3 Random[]],
                    MapIndexed[If[(-1)^(k+l) == (-1)^Plus@@#2, #, {}] &,
                       ExtendedRiemannSpherePatch[
           Function[{x, y}, Abs[Exp[(Log[x + i y] + 2 k π i)
               Exp[(Log[x + i y] + 2 l π i) (x + i y)]]]], {0, π},
           {π / 2, π}, {80, 50}], {2}]}, {k, -1, 1}, {l, -1, 1}]],
      Boxed → False, ViewPoint → {3, -3, 0.8}];
```

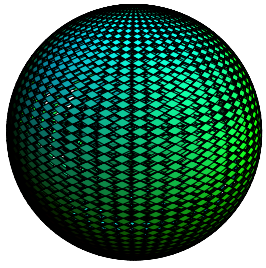
From In[76]:=



Here we show some sheets of the function $w(z) = \arcsin(z)$.

```
In[77]:= pic = Show[Graphics3D[{EdgeForm[], GrayLevel[0],
           Thickness[0.0001], SurfaceColor[Hue[0.32], Hue[0.58], 1.92],
             Table[{ExtendedRiemannSpherePatch[Function[{x, y},
               Re[ArcSin[x + i y] + 2 π k]], {0, 2 π}, {0, π}, {50, 50}],
                        ExtendedRiemannSpherePatch[
              Function[{x, y}, π - Re[ArcSin[x + i y] + 2 π k]],
              {0, 2 π}, {0, π}, {50, 50}]}, {k, -2, 2}] /.
            p_Polygon :→ holePolygon[p, 1, NewPolygonShape → CutVertices]}],
       Boxed → False, ViewPoint → {2, -3, 0.8}];
```

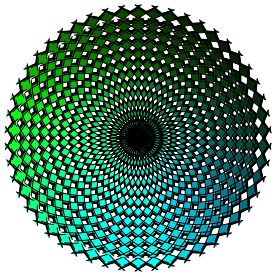
From In[77]:=



To see the three branch points more clearly, we cut the sphere in half along the x-y-plane, the x-z-plane, and the y-z-plane.

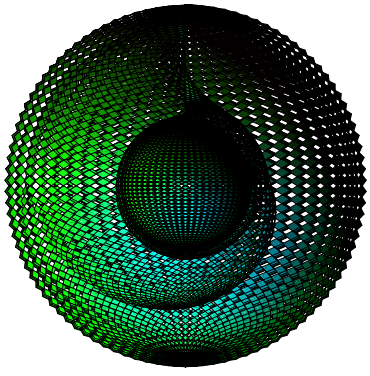
In[78]:=
```
Show[pic, ViewPoint → {0, 0, 2}, PlotRange → {All, All, {-1.5, 0}}];
```

From In[78]:=



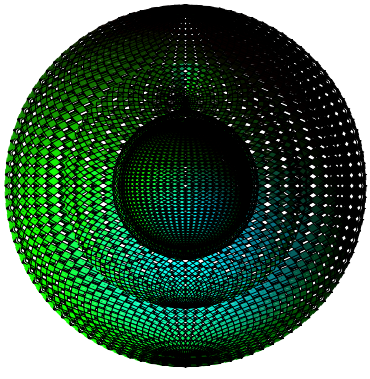
In[79]:=
```
Show[pic, ViewPoint → {0, 2, 0}, PlotRange → {All, {-1.5, 0}, All}];
```

From In[79]:=



In[80]:=
```
Show[pic, ViewPoint → {2, 0, 0}, PlotRange → {{-1.5, 0}, All, All}];
```

From In[80]:=

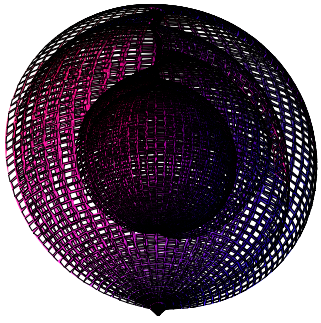


In[81]:=
```
Remove[pic]; (* to save memory *)
```

The function $w(z) = \sqrt{\ln(z)}$ has the two branch points 0 and 1. They are more visible in the following half of the generalized Riemann sphere.

In[82]:=
```
Show[Graphics3D[{EdgeForm[], Thickness[0.0001],
  GrayLevel[0], SurfaceColor[Hue[0.73], Hue[0.02], 1.94],
    Table[ExtendedRiemannSpherePatch[
   Function[{x, y}, Im[(-1)^j Sqrt[Log[x + i y] + 2 π k i]]], {0, π},
   {0, π}, {24, 60}], {k, -1, 1}, {j, 0, 1}]} /. p_Polygon :>
  holePolygon[p, 0.6, NewPolygonShape → RandomQuadrilateral]],
 Boxed → False, ViewPoint → {0.7, -3, 1.1}];
```

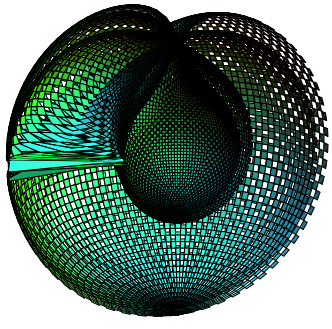
From In[82]:=



The function $w(z) = \frac{\ln(z+1)}{-z-1} + \frac{\ln(-z-1)}{z+1}$ has the two branch points $z = \pm 1$. At these points we have an additional pole.

In[83]:=
```
Show[Graphics3D[{EdgeForm[{GrayLevel[0], Thickness[0.0001]}],
 SurfaceColor[Hue[Random[]], Hue[Random[]], 3 Random[]],
    Table[MapIndexed[If[(-1)^(k+1) == (-1)^(Plus@@#2), #, {}] &,
      ExtendedRiemannSpherePatch[Function[{x, y},
       Im[(Log[x + i y + 1] + 2 π k i)/(-(x + i y) - 1) +
        (Log[-(x + i y) - 1] + 2 π l i)/(x + i y + 1)]],
       {0, π}, {0, π}, {60, 60}], {2}], {k, -1, 1}, {l, -1, 1}]}],
 Boxed → False, ViewPoint → {-1, -3, 0.8}];
```

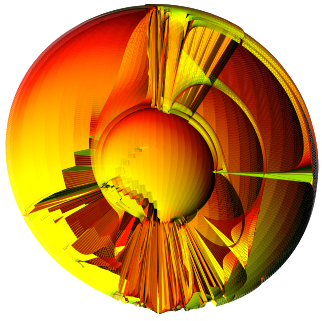
From In[83]:=



The next picture shows some sheets of the function $w(z) = \ln\left(\sin\left(\sqrt{z} + \frac{1}{z}\right)\right)$. Now the north pole (as well as the south pole) shows the accumulation points of logarithmic branch points. We exclude points near the poles to avoid resolution problems.

```
In[84]:= Show[Graphics3D[{EdgeForm[],
         GrayLevel[0], SurfaceColor[Hue[0.04], Hue[0.26], 1.08],
             Table[ExtendedRiemannSpherePatch[
           Function[{x, y}, (Re[#] + Im[#]) &[
             Log[Sin[(-1)^j Sqrt[x + i y] + 1 / (x + i y)]] + 2 π k i]],
           {0, π}, {10^-2, π - 10^-2}, {24, 180}], {k, -1, 1}, {j, 0, 1}]}],
       Boxed → False, ViewPoint → {0.7, -3, 1.1}];
```

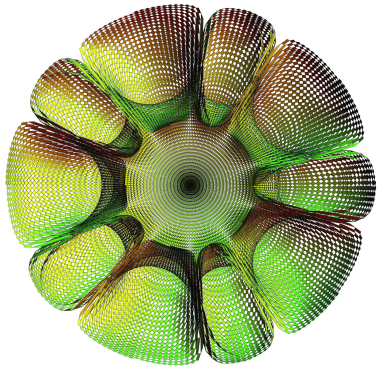
From In[84]:=



The next picture shows some sheets of the function $\ln^2(z^5 - 1)$. On the unit circle there are five branch points (and branch cuts going out from them) and five additional branch cuts between them.

```
In[85]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001], GrayLevel[0],
         GrayLevel[0], SurfaceColor[Hue[0.22], Hue[0.28], 1.85],
         Map[Table[ExtendedRiemannSpherePatch[
               Function[{x, y}, Im[(Log[(x + i y)^5 - 1] + 2 k π i)^2]], #,
               {π / 2, π}, {20, 48}], {k, -2, 2}] &, {10^-6, -10^-6} + # &/@
           Partition[Table[φ, {φ, 0, 2 π, 2 π / 10}], 2, 1], {1}] /.
        p_Polygon :> holePolygon[p, 1, NewPolygonShape → CutVertices]}],
       Boxed → False, ViewPoint → {0, 0, 2.1}];
```

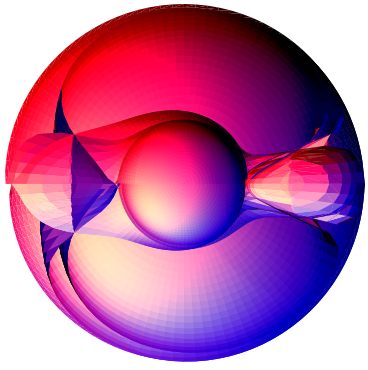
From In[85]:=



The next picture shows the function $\ln^2\left(z-1+\sqrt[3]{z+1}\right)$. The branch points are 0 and 1.

```
In[86]:= Show[Graphics3D[{EdgeForm[], Thickness[0.0001], GrayLevel[0],
           GrayLevel[0], SurfaceColor[Hue[0.85], Hue[0.15], 1.78],
           Map[    Table[ExtendedRiemannSpherePatch[Function[{x, y},
                Im[(Log[x + i y - 1] + 2 k π i + (-1)^j (x + i y + 1)^(1/3))^2]], #, {π/2, π},
              {48, 24}], {k, -1, 1}, {j, -1, 1}] &, {10^-6, -10^-6} + # & /@
            Partition[Table[φ, {φ, 0, 2 π, 2 π/2}], 2, 1], {1}]}],
         Boxed → False, ViewPoint → {0, 0, 2.1}];
```

From In[86]:=



One could now go on and visualize more complicated Riemann surfaces on the Riemann sphere. In the spirit of the last section, we would rewrite **oder** as a differential equation **ode**θ for the longitudinal coordinate of a spherical coordinate θ system according to the following formula, together with corresponding modifications in **patch** and **RiemannSurface**.

```
In[87]:= odeθ[oder_, {ψ_, ρ_}, {ϕ_, θ_}] := oder //.
           {ψ'[ρ] == rhs_ → ϕ'[θ] == (Cos[θ] - 1) rhs, ψ[ρ] → ϕ[θ], ρ → Cot[θ/2]}
```

Such an extension is straightforward; we end now and leave further explorations in this direction to the interested reader.

Demystifying Rules

Nancy Blachman

President
Variable Symbols, Inc.
356 Bush Street
Mountain View, CA 94041
Fax: 650-966-8998
Tel: 650-966-8999
Nancy@VariableSymbols.com
www.VariableSymbols.com

Have you ever wondered in what order *Mathematica* applies replacement rules or why *Mathematica* sometimes applies a replacement rule and other times doesn't? This article explains how replacement rules work and why they are useful.

■ Replacement Rules

Rules are a mechanism for replacing a variable with a value in an expression. The **ReplaceAll** operator, which is abbreviated **/.** (with no space between the **/** and the **.**), applies a rule or list of rules after evaluating the expression. Think of **/.** as meaning *given that* and **->** (a minus sign together with a greater than sign, with no space in between the two) as meaning *goes to*. So the expression **2x+3y/. x->5b** means **2x+3y** *given that* **x** *goes to* **5b**.

```
In[1]:= 2 x + 3 y /. x -> 5 b

Out[1]= 10 b + 3 y
```

Notice that all **x**'s in the expression to the left of **/.** are replaced by **5b**, but that **x** has not been assigned a value.

```
In[2]:= x

Out[2]= x
```

Here are some more examples of replacement rules, each followed by an explanation of the replacement.

```
In[3]:= 2 x + 3 y /. {x -> 7, 3 y -> 5 a}

Out[3]= 14 + 5 a
```

The symbol **x** gets replaced by **7** and the expression **3y** gets replaced by **5a**, so the expression **2x+3y** is transformed into **2*7+5a**, which evaluates to **14+5a**.

 © 2002 Wolfram Media, Inc.

```
In[4]:= 2 x + 3 y /. {x -> a, 2 x -> b}

Out[4]= b + 3 y
```

Mathematica uses a greedy algorithm. It looks for the largest expression that matches a pattern. First *Mathematica* checks whether there is a rule that matches the entire expression. If not, it checks whether there is a rule that matches part of an expression. In this case, the rule **2x->b** matches part of the expression **2x+3y**, so the rule is applied. Another way to look at this is that *Mathematica* matches the most specific expression, and the rule **2x->b** is more specific than the rule **x->a**.

```
In[5]:= 2 x + 3 y /. {x -> a, x -> b}

Out[5]= 2 a + 3 y
```

Rules of equal specificity are applied from left to right, so the rule **x->a** is applied before **x->b**.

```
In[6]:= 2 x + 3 y /. {x -> b, x -> a}

Out[6]= 2 b + 3 y
```

If we put the rule **x->b** first, it is applied before **x->a**.

```
In[7]:= 2 x + 3 y /. {x -> b, 2 x -> a}

Out[7]= a + 3 y
```

The rule **2x->a** is applied here because it matches a larger piece of the entire expression than the rule **x->b**.

```
In[8]:= 2 x + 3 y /. {{x -> a}, {x -> b}}

Out[8]= {2 a + 3 y, 2 b + 3 y}
```

When *Mathematica* is given a list of lists of rules, each sublist is applied to the original expression. The result is a list of transformed expressions.

```
In[9]:= 1 / Sqrt[x^2] /. Sqrt[x^2] -> 9

Out[9]=  1
         ------
          2
         Sqrt[x ]
```

Since *Mathematica* doesn't know anything about **x^2** (whether it is positive or negative), it leaves the expression **Sqrt[x^2]** unchanged. *Mathematica* internally represents **1/Sqrt[x^2]** as **(x^2)^-(1/2)** and **Sqrt[x^2]** as **(x^2)^(1/2)**.

```
In[10]:= FullForm[1 / Sqrt[x^2]]

Out[10]//FullForm=
         Power[Power[x, 2], Rational[-1, 2]]

In[11]:= FullForm[Sqrt[x^2]]

Out[11]//FullForm=
         Power[Power[x, 2], Rational[1, 2]]
```

Consequently, *Mathematica* can't find the expression **Sqrt[x^2]** in the original expression **1/Sqrt[x^2]**. Therefore, the rule isn't applied and *Mathematica* returns the original expression unchanged.

```
In[12]:= Sqrt[x] + 1 / Sqrt[x] /. {Sqrt[x] -> a, x -> b}
```

$$\text{Out[12]= } a + \frac{1}{\sqrt{b}}$$

Because **Sqrt[x]** doesn't appear in the expression **1/Sqrt[x]**, the first rule (**Sqrt[x]->a**) is only applied to the first element of the sum (**Sqrt[x]**). The second rule (**x->b**) is applied to the second element (**1/Sqrt[x]**).

```
In[13]:= {1, 11, 111, 1111} /. {1 -> one}
Out[13]= {one, 11, 111, 1111}
```

Numbers are atomic expressions and can't be subdivided. The rule **1 -> one** is applied only when **1** matches the entire number.

```
In[14]:= 2 x + 3 x /. {2 x -> y, 3 x -> 3}
Out[14]= 5 x
```

Mathematica evaluates the expression to the left of the **/.** before applying any of the rules. The expression **2x+3x** evaluates to **5x** and then none of the rules match, so they aren't applied.

```
In[15]:= 2 x + 3 y /. {2 x -> a, a -> b}
Out[15]= a + 3 y
```

Mathematica applies only those rules that apply to the original expression, so **2x** gets replaced by **a**. No other rule applies to the original expression.

```
In[16]:= 2 x + 3 y //. {2 x -> a, a -> b}
Out[16]= b + 3 y
```

The notation **//.** is an alias for **ReplaceRepeated**. The rules are applied repeatedly until there are no more rules that apply. On the first pass, **2x** gets replaced by **a**. On the second pass, **a** gets replaced by **b**.

```
In[17]:= {a, b, c, d} /. {a -> b, b -> c, c -> b, d -> a}
Out[17]= {b, c, b, a}
```

The symbol **a** gets replaced by **b**, the symbol **b** gets replaced by **c**, and the symbol **c** gets replaced by **a**.

```
In[18]:= {a, b, c, d} //. {a -> b, b -> c, c -> d, d -> a}
ReplaceRepeated::rrlim : Exiting after {a, b, c, d} scanned 65536 times.
Out[18]= {a, b, c, d}
```

The rules are applied repeatedly. After applying the result many, many times, *Mathematica* stops so that it won't get caught in an infinite loop.

■ Using Rules

Let's look at some examples of functions that return results in terms of rules. **Solve** finds the solutions to a polynomial equation of degree four or less (as well as some polynomial equations of higher degree). The solution is stated as a list of replacement rules.

```
In[19]:= soln = Solve[x^3 - 3 x^2 - 17 x + 51 == 0, x]
```

$$\text{Out[19]= } \left\{\{x \to 3\}, \left\{x \to -\sqrt{17}\right\}, \left\{x \to \sqrt{17}\right\}\right\}$$

We can get a list of the values for **x** by using **ReplaceAll**.

```
In[20]:= x /. soln
```

$$\text{Out[20]= } \left\{3, -\sqrt{17}, \sqrt{17}\right\}$$

FindRoot, **DSolve**, and **NSolve** also return solutions in terms of replacement rules.

```
In[21]:= FindRoot[Sin[x] / x, {x, 3, 4}]
```

$$\text{Out[21]= } \{x \to 3.14159\}$$

```
In[22]:= DSolve[y''[x] + 2 y'[x] == 0, y[x], x]
```

$$\text{Out[22]= } \left\{\left\{y[x] \to -\frac{1}{2} e^{-2x} C[1] + C[2]\right\}\right\}$$

```
In[23]:= p = x^10 + x^9 - x^7 - x^6 - x^5 - x^4 - x^3 + x + 1;

In[24]:= complexRoots = NSolve[p == 0]
```

```
Out[24]= {{x → -0.943305 - 0.331928 i},
  {x → -0.943305 + 0.331928 i}, {x → -0.734438 - 0.678676 i},
  {x → -0.734438 + 0.678676 i}, {x → -0.292332 - 0.956317 i},
  {x → -0.292332 + 0.956317 i}, {x → 0.456866 - 0.889536 i},
  {x → 0.456866 + 0.889536 i}, {x → 0.850137}, {x → 1.17628}}
```

We can easily compute the absolute values of the roots.

```
In[25]:= Abs[x] /. complexRoots
```

```
Out[25]= {1., 1., 1., 1., 1., 1., 1., 1., 0.850137, 1.17628}
```

What do the absolute values of the complex roots have in common?

The function **DispersionReport**, from the standard package **Statistics`DescriptiveStatistics`**, returns a list of statistics in the form of rules.

```
In[26]:= Needs["Statistics`DescriptiveStatistics`"]

In[27]:= dReport = DispersionReport[{3, 7, 2, 3, 1, 2, 8}]
```

$$\text{Out[27]= } \left\{\text{Variance} \to \frac{152}{21}, \text{StandardDeviation} \to 2\sqrt{\frac{38}{21}}, \text{SampleRange} \to 7,\right.$$
$$\left.\text{MeanDeviation} \to \frac{106}{49}, \text{MedianDeviation} \to 1, \text{QuartileDeviation} \to 2\right\}$$

We can pick out a particular result or several results from the list.

```
In[28]:= MeanDeviation /. dReport
```

$$\text{Out[28]}= \frac{106}{49}$$

```
In[29]:= {Variance, StandardDeviation} /. dReport
```

$$\text{Out[29]}= \left\{\frac{152}{21}, 2\sqrt{\frac{38}{21}}\right\}$$

Graphics options are specified in terms of rules.

```
In[30]:= Options[Plot]
```

```
Out[30]= {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
  ColorOutput → Automatic, Compiled → True, DefaultColor → Automatic,
  Epilog → {}, Frame → False, FrameLabel → None, FrameStyle → Automatic,
  FrameTicks → Automatic, GridLines → None, ImageSize → Automatic,
  MaxBend → 10., PlotDivision → 30., PlotLabel → None, PlotPoints → 25,
  PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
  Prolog → {}, RotateLabel → True, Ticks → Automatic,
  DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction,
  FormatType :→ $FormatType, TextStyle :→ $TextStyle}
```

```
In[31]:= PlotPoints /. Options[Plot]
```

```
Out[31]= 25
```

■ Naming Parts of Expressions

Rules provide a mechanism for assigning names to parts of an expressions and then specifying what to do with some or all the names.

```
In[32]:= nationalDebt =
           {{1900, 1263}, {1930, 16185}, {1960, 284093}, {1990, 3233313}};
```

```
In[33]:= ListPlot[nationalDebt, PlotJoined -> True];
```

```
From In[33]:=
```



The notation **year_** is a named pattern that designates a *Mathematica* expression. The rule {**year_,debt_->{year,Log[debt]}** specifies that given a pair of

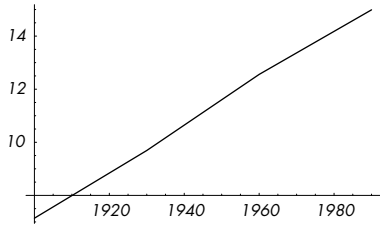
values, *Mathematica* assigns the name **year** to the first element and **debt** to the second element in the pair, and returns the pair **{year,Log[debt]}**.

```
In[34]:= nationalDebt /. {year_, debt_} -> {year, Log[debt]}

Out[34]= {{1900, Log[1263]}, {1930, Log[16185]},
          {1960, Log[284093]}, {1990, Log[3233313]}}

In[35]:= ListPlot[%, PlotJoined -> True];
```

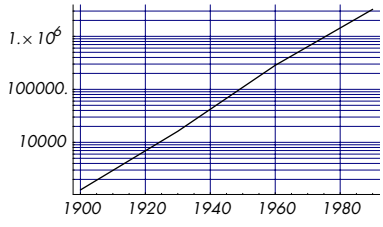
From In[35]:=



Actually, we didn't need to use this rule because there's a function for making log plots.

```
In[36]:= Needs["Graphics`Graphics`"];
         LogListPlot[nationalDebt,
           PlotJoined -> True, GridLines -> Automatic];
```

From In[36]:=



Let's look at some more examples of rules with named patterns.

```
In[38]:= {{1, 2, 3}, {4, 5, 6}} /. {x_, y_, z_} -> {x, Log[y]}

Out[38]= {{1, Log[2]}, {4, Log[5]}}
```

Triples are replaced by pairs.

```
In[39]:= {{1, 2}, {3, 4}, {5, 6}} /. {x_, y_} -> {y, x}

Out[39]= {{2, 1}, {4, 3}, {6, 5}}
```

The rule swaps the elements in each pair.

```
In[40]:= {{1, 2, 3}, {4, 5, 6}} /. {x_, y_} -> {y, x}

Out[40]= {{4, 5, 6}, {1, 2, 3}}
```

The elements in a pair are swapped. There is only one pair in the expression.

```
In[41]:= {{1, 2}, {3, 4}, {5, 6}} /. {x_, y_} -> {x, Log[y]}

Out[41]= {{1, Log[2]}, {3, Log[4]}, {5, Log[6]}}
```

Each pair gets replaced by another pair consisting of the first element and the log of the second element.

```
In[42]:= {{1, 2}, {3, 4}} /. {x_, y_} -> {x, Log[y]}

Out[42]= {{1, 2}, {Log[3], Log[4]}}
```

The largest (outermost) pair is replaced by another pair consisting of the first element and the log of the second element.

```
In[43]:= f[a + f[b]] /. f -> g

Out[43]= g[a + g[b]]
```

The symbol **f** is replaced by **g**.

```
In[44]:= f[a + f[b]] /. f[x_] -> g[x]

Out[44]= g[a + f[b]]
```

Since the rule matches the entire expression, *Mathematica* replaces the head of the expression with **g**. It doesn't look deeper into the expression.

```
In[45]:= f[a + f[b]] //. f[x_] -> g[x]

Out[45]= g[a + g[b]]
```

Mathematica repeatedly replaces **f** with **g** until the rule no longer applies.

```
In[46]:= Log[x] + Log[y + Log[z + Log[t]]] /. Log[w_] -> w

Out[46]= x + y + Log[z + Log[t]]
```

Mathematica applies the rule to the subexpressions **Log[x]** and **Log[y + Log[t]]** and transforms them to **x** and **y+Log[z+Log[t]]**. The rule doesn't get applied to subexpressions of these expressions. Once a transformation is made, *Mathematica* doesn't apply the rule again, unless we use **//.** or **ReplaceRepeated**. So, the rule doesn't get applied to the nested calls to **Log**.

```
In[47]:= Log[x] + Log[y + Log[z + Log[t]]] //. Log[w_] -> w

Out[47]= t + x + y + z
```

■ Delayed Rules

The notation **->** is the alias for the function **Rule**. The notation **:>** is the alias for the function **RuleDelayed**. The notation **x->Random[]** specifies a rule to replace **x** with the evaluated **Random[]**. The expression **Random[]** evaluates to a random number. The notation **x:>Random[]** specifies a rule to replace **x** with the unevaluated expression **Random[]**.

```
In[48]:= Table[x, {3}]

Out[48]= {x, x, x}
```

```
In[49]:= Table[x, {3}] /. x -> Random[]

Out[49]= {0.810595, 0.810595, 0.810595}

In[50]:= Table[x, {3}] /. x :> Random[]

Out[50]= {0.0966423, 0.243373, 0.902881}
```

Delayed rules are like delayed assignments in that the right side of the rule is evaluated after the replacement has been made.

Using **Trace**, you can see the order in which *Mathematica* evaluates these expressions. With an immediate assignment, *Mathematica* first replaces **Table** with a list of three **x**'s. Then *Mathematica* calls **Random** and obtains a value. Then each **x** in the list is replaced by that value.

```
In[51]:= Table[x, {3}] /. x -> Random[] // Trace // ColumnForm

Out[51]= {Table[x, {3}], {x, x, x}}
         {{Random[], 0.0153378}, x → 0.0153378, x → 0.0153378}
         {x, x, x} /. x → 0.0153378
         {0.0153378, 0.0153378, 0.0153378}
```

With a delayed assignment, *Mathematica* first replaces **Table** with a list of three **x**'s. Then *Mathematica* replaces each **x** in the list with a call to **Random**. Then *Mathematica* evaluates each call to **Random**.

```
In[52]:= Table[x, {3}] /. x :> Random[] // Trace // ColumnForm

Out[52]= {Table[x, {3}], {x, x, x}}
         {x :→ Random[], x :→ Random[]}
         {x, x, x} /. x :→ Random[]
         {Random[], Random[], Random[]}
         {Random[], 0.0214599}
         {Random[], 0.137507}
         {Random[], 0.494486}
         {0.0214599, 0.137507, 0.494486}
```

You can specify rules to transform sums of angles. These rules used to be in the standard package **Algebra`Trigonometry`**. In Version 3, they were incorporated into the *Mathematica* kernel.

```
In[53]:= sumsOfAngleRules = {
          Sin[x_ + y_] :> Sin[x] Cos[y] + Sin[y] Cos[x],
          Cos[x_ + y_] :> Cos[x] Cos[y] - Sin[x] Sin[y],
          Tan[x_ + y_] :> (Tan[x] + Tan[y]) / (1 - Tan[x] Tan[y]) }

Out[53]= {Sin[x_ + y_] :→ Sin[x] Cos[y] + Sin[y] Cos[x],
          Cos[x_ + y_] :→ Cos[x] Cos[y] - Sin[x] Sin[y],
          Tan[x_ + y_] :→ (Tan[x] + Tan[y]) / (1 - Tan[x] Tan[y])}
```

■ Exercises

I hope you now understand how rules work and how you can use them to manipulate your expressions. Here are some exercises to test your understanding.

1. For the expression

```
In[54]:= a = (x + y)^2 Sin[2 (x + 3 y)];
```

write a rule that expands the argument of **Sin**, but does not expand the term **(x+y)^2**.

Solution:

```
In[55]:= a /. Sin[x_] :> Sin[Expand[x]]
Out[55]= (x + y)^2 Sin[2 x + 6 y]
```

2. In the substitution

```
In[56]:= y (x + 2) (x - 2) /. y z_ -> y Expand[z]
Out[56]= (-2 + x) (2 + x) y
```

why doesn't the rule expand the term **(x+2)(x-2)**? Write a rule that will expand **y(x+2)(x-2)**.

Solution: The right-hand side of the rule gets evaluated immediately, so **Expand[z]** evaluates to **z**. Use a delayed rule instead.

```
In[57]:= y (x + 2) (x - 2) /. y z_ :> y Expand[z]
Out[57]= (-4 + x^2) y
```

3. Given five points $\{\{x_1, y_1\}, \ldots, \{x_5, y_5\}\}$ and five heights $\{z_1, \ldots, z_5\}$ at those points, construct triplets of the form $\{\{x_1, y_1, z_1\}, \ldots, \{x_5, y_5, z_5\}\}$ using only list manipulation constructs and rules, and without using **Part** or functions with iterators (such as **Table** or **Do**).

Solution:

```
In[58]:= xyPairs = Transpose[{Array[x, {5}], Array[y, {5}]}]
Out[58]= {{x[1], y[1]}, {x[2], y[2]}, {x[3], y[3]}, {x[4], y[4]}, {x[5], y[5]}}
In[59]:= zValues = Array[z, {5}]
Out[59]= {z[1], z[2], z[3], z[4], z[5]}
In[60]:= Transpose[{xyPairs, zValues}] /. {{x_, y_}, z_} -> {x, y, z}
Out[60]= {{x[1], y[1], z[1]}, {x[2], y[2], z[2]},
          {x[3], y[3], z[3]}, {x[4], y[4], z[4]}, {x[5], y[5], z[5]}}
```

4. **FactorInteger** returns a list of the prime factors of its argument together with their exponents. For example,

```
In[61]:= FactorInteger[909]
Out[61]= {{3, 2}, {101, 1}}
```

If you raise the first number in the pair to the power of the second number and then multiply the results together, you should obtain the number you factored.

```
In[62]:= 3^2 101^1
Out[62]= 909
```

Write a rule that takes the result returned by **FactorInteger** and computes the original number.

Solution:

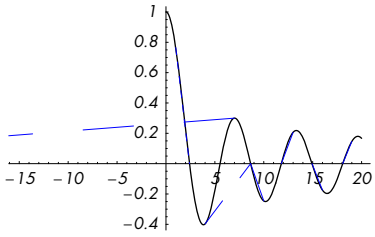
```
In[63]:= factors = FactorInteger[90909]
Out[63]= {{3, 3}, {7, 1}, {13, 1}, {37, 1}}
In[64]:= factors /. {f_Integer, e_} -> f^e
Out[64]= {27, 7, 13, 37}
In[65]:= Apply[Times, %]
Out[65]= 90909
In[66]:= FactorInteger[9090909] /. {f_Integer, e_} -> f^e //
          Apply[Times, #] &
Out[66]= 9090909
```

5. **FindRoot** uses Newton's method to obtain a root, given a function and a starting value. Plot the function **BesselJ[0,x]** for x in the range [0, 20]. Choose seven starting values of x in that range, and for each one, draw the line segment that joins the point **{x,BesselJ[0,x]}** on the graph to the point **{r,0}** on the x-axis, where **r** is the root returned by **FindRoot** when given the starting value **x**. (You can combine a graph of a function with a line produced with the command **Graphics** by using **Show**.)

Solution:

```
In[67]:= Show[Plot[BesselJ[0, x], {x, 0, 20}, DisplayFunction -> Identity],
     Graphics[{Thickness[0.05], RGBColor[0, 0, 1],
            Table[{Thickness[0.002],
       Dashing[{x0 / 50.}], Line[{{x0, N[BesselJ[0, x0]]},
         { x /. FindRoot[BesselJ[0, x], {x, x0}], 0}}]},
      {x0, 1, 19, 3}]
         }],
     DisplayFunction :> $DisplayFunction];
```

From In[67]:=



About the Author

Nancy Blachman is president and founder of Variable Symbols, Inc., a company that specializes in training and consulting in *Mathematica* and Linux. She is author of the popular tutorial book *Mathematica: A Practical Approach* (Prentice Hall). From 1987–1989, she taught "Problem Solving with *Mathematica*" at Stanford University.

Animated Three-Dimensional Structure of the Earth's Interior for Research and Education

Hiroki Sato, Kenichi Muro

Department of Earth and Space Science, Osaka University, Toyonaka, Osaka 560-0043, Japan
roki@ess.sci.osaka-u.ac.jp

Akira Hasegawa

Research Center for Prediction of Earthquakes and Volcanic Eruptions, Tohoku University, Sendai 980-0845, Japan

Seismic tomographic studies have determined three-dimensional (3D) velocity structures, in detail, of the crust and the upper mantle of the Earth. Yet, simple two-dimensional (2D) sections have generally been used to present 3D tomographic results. Here we show 3D views and animations of the Earth's structure that are made as easy as 2D sections by using *Mathematica*. As an example, low-velocity zones in the upper mantle are shown in three dimensions together with major volcanoes, mid-crustal reflectors, earthquake hypocenters, the Moho discontinuity, and the upper plane of the subducted slab that are observed in northeastern Japan. Low velocities in volcanic areas generally correspond to high temperature and indicate possible presence of magma. The 3D animations enable us to investigate the special correlation between low-velocity zones, volcanoes, reflectors, earthquakes, and the slab, and thus enable us to study magma ascent pathways in detail.

 © 2002 Wolfram Media, Inc.

■ Introduction

Earthquakes generate seismic waves of compressional and shear modes (P and S waves, respectively). Much of the knowledge about the internal structure of the Earth comes from earthquake observations, as seismic waves propagate throughout the globe. Many seismic stations on the surface of the Earth together yield a large amount of seismic data, which enable observers to determine accurate locations of earthquake hypocenters and accurate travel time of seismic waves. Seismologists thus determine velocities of seismic waves propagating through the Earth. Generally, waves travel faster through cold material than they do through hot. Temperature heterogeneity exists in the Earth because cold lithosphere (plate) subducts at the trench and high-temperature lava erupts at volcanoes. Therefore, high- or low-seismic velocity regions are present within the Earth, and differences from the average velocity (called *velocity perturbations*) are determined from seismic studies. An image of velocity perturbations of the inner Earth is called *seismic tomography*. Analogous to X-rays in medical tomography (CT: computerized tomography), seismic rays are used to probe the Earth in seismic tomography. Differences in velocity allow scientists to calculate the size, density, and elastic properties of the Earth's interior and use that information to make predictions about the shape, materials, and thermal conditions of the Earth. The knowledge of the internal structure is essential to understand geologic processes like plate tectonics and continental drift.

Recently, major advances in the study of seismic tomography have been made with the advent of computer software and hardware. One of them is the development of three-dimensional (3D) seismic tomography, which unveils many important aspects of the Earth. In general, however, many 3D tomographic results have been shown in two-dimensional (2D) sections. 3D presentations are only rarely made. In 2D sections, it is not easy to investigate, for example, the spatial correlation between low- (or high-) velocity regions and important geophysical processes (for example, earthquake hypocenter, volcano location, and plate movement). Such correlation, however, is clearly seen and studied in detail in a 3D view.

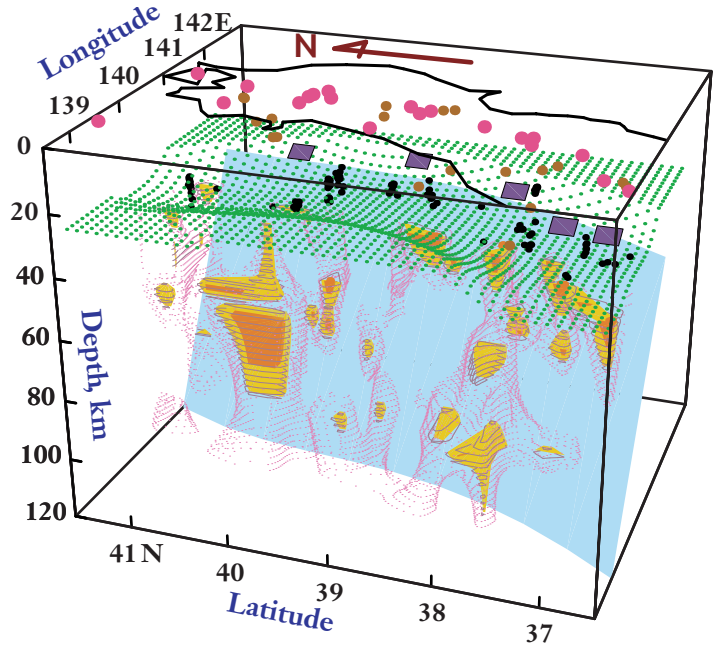
In this paper, we present a simple, practical, and effective way to show a 3D view and animation of the Earth's structure by using *Mathematica*. We select the region of northeastern (NE) Japan where interesting features typical to a volcanic area are observed. They are the presence of mid-crustal S wave reflectors and the occurrence of low-frequency microearthquakes [1, 2]. The reflectors are thin magma bodies located at about 8 to 15 km beneath the surface. Due to the acoustic impedance difference between magma and surrounding crustal rocks like granite, S waves are distinctly reflected on the magma-rock interface. Microearthquakes of magnitude less than 2.5 are frequently observed in and around the low-velocity zones, and have relatively low-characteristic frequencies (around 2 Hz), compared with the ordinary seismic frequency band (up to ~20 Hz). These observations indicate the close relationship between magnetism, slow-velocity anomalies, low-frequency microearthquakes and S wave reflectors [2, 3]. Here, the low-velocity zones are shown in three dimensions, and are discussed in relation to the occurrence of low-frequency microearthquakes and the presence

of mid-crustal S wave reflectors. The 3D views suggest magma ascent pathways in the Japan island arc.

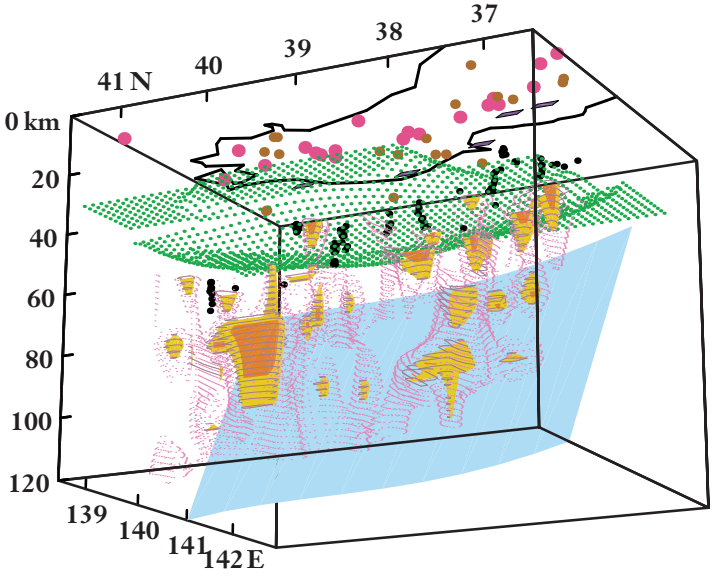
■ 3D Structure and Animation

As the cold Pacific plate subducts beneath NE Japan, high-temperature magma is produced and causes an eruption of an island arc volcano. Many observed seismic activities, either large- or microearthquakes, have led seismologists to determine the detailed velocity structure beneath Japan [2, 4]. For example, Zhao and others [4] analyzed 14,045 arrival times including both P and S waves. Complex seismic velocity discontinuities (the Moho discontinuity and the upper plane of the subducting Pacific plate) are taken into account in their 3D velocity inversion. Their results have shown low-velocity zones in the crust beneath the volcanoes and in the central part of the mantle wedge, and up to 6 percent higher velocity in the Pacific plate. Their tomography results, given in FORTRAN code, yield a value of velocity perturbation in percentages at any point of the upper mantle beneath NE Japan. For a 3D animation, here we use their 3D P wave velocity structure.

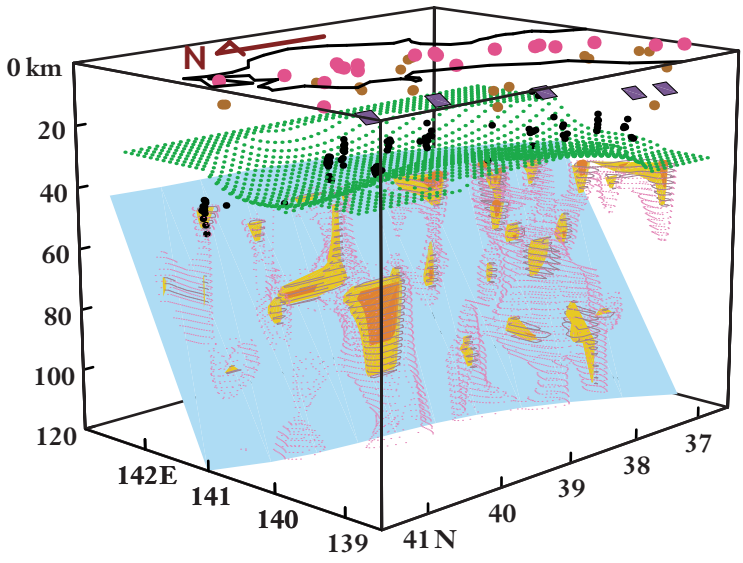
Figures 1 (a) to (e) show a 3D mapping of the Earth's interior to 120 km depth beneath NE Japan (lat. 36.5-41.6 N, long. 138.5-142.5 E) based on the P wave seismic tomography by Zhao and others [4]. In this figure, the vertical scale is exaggerated 3.2 times so that the details of the internal structure can be seen clearly. Programs are given in Listings 1 to 3 for 3D views and animations. Even simple *Mathematica* code yields 3D Earth structure in considerable detail. At the top of the figures (the Earth's surface), the location of active volcanoes (red circles) and the map of NE Japan are shown. To be transparent, the Moho discontinuity and the surface of –3% velocity perturbation are described, not by planes, but by green and pink points, respectively. The Moho discontinuity is a boundary between the crust and the mantle. The upper plane of the subducted Pacific plate is shown by a combination of ten light blue rectangles. In Listings 1 to 3, dv-3 is a set of points of –3% velocity perturbation, while dv-4 and dv-5 consist of the data tracing outlines of the north-south cross sections of –4% and –5% velocity perturbations. These low-velocity regions are shown in polygons (yellow and orange, respectively) with contour lines that reduce the data quite a lot (compared with points) and help to save memory. Within the yellow and orange polygons, velocity perturbations are less than –4% and –5%, respectively. The contour lines (light purple, the data file dv-4c) trace outlines of –4% velocity perturbation at every 1 km depth. These low-velocity data can be shown in dots, too; however, the memory needed for 3D views and animations becomes quite large.


Longitude
142E
141
140
139
N
0
20
40
60
80
100
120
Depth, km
41 N
40
39
38
37
Latitude

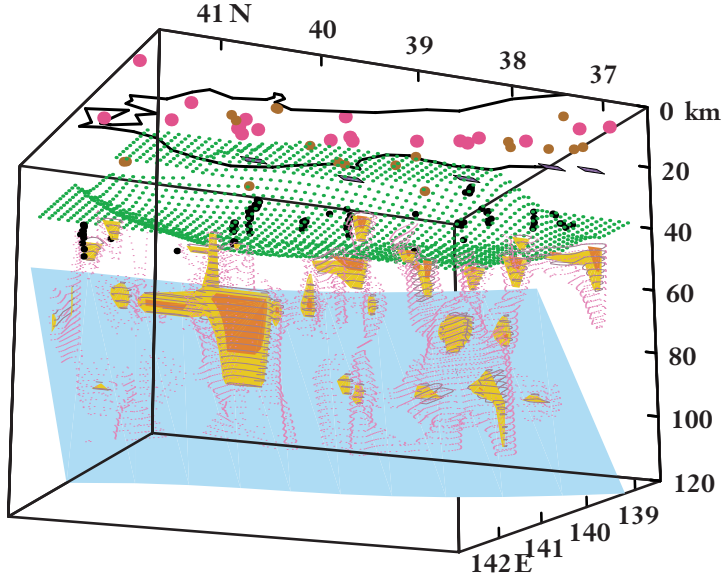
(a)


37
38
39
40
41 N
0 km
20
40
60
80
100
120
139
140
141
142 E

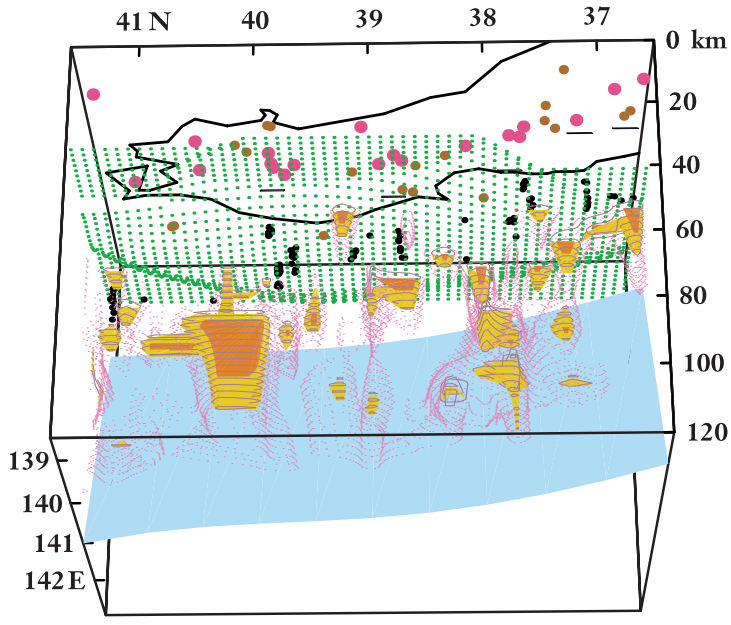
(b)


N
0 km
20
40
60
80
100
120
142E
141
140
139
41 N
40
39
38
37

(c)


41 N
40
39
38
37
0 km
20
40
60
80
100
120
139
140
141
142 E

(d)



(e)

- active volcanoes
- large crustal earthquakes (M≥6)
- low-frequency microearthquakes
- mid-crustal S wave reflectors
- Moho discontinuity
- upper plane of subducted slab
- less than –4% velocity perturbation
- less than –5% velocity perturbation
- contours of –3% velocity perturbation

To see the various elements in color, please refer to the electronic version.

Figure 1. Three-dimensional mapping of low-velocity zones and seismic activities in northeastern Japan as viewed from (a) above and (b) below from southwest direction, from (c) above and (d) below from northwest direction, and from (e) below from west (symbols are described in (e)). Vertical scale is exaggerated 3.2 times. It is observed that fairly low-velocity zones exist locally beneath the sites of volcanoes and downwards to the west. The high temperature regions (regions of less than –3% velocity perturbation) appear to extend to greater depths in the mantle wedge, suggesting a plume-like uprising of hot mantle rocks. In between the volcanoes and the low-velocity zones, S wave reflectors exist and low-frequency microearthquakes occur. Two major low-velocity zones are observed below 60 km depth and branch off at the uppermost mantle, and eventually underplate the Moho right beneath the volcanoes.

```
dv1={PointSize[0.001],RGBColor[0.9,0.4,1],Map[Point,<<dv-3]}
dv2={EdgeForm[],FaceForm[RGBColor[0.9,0.8,0],
   RGBColor[0.9,0.8,0]],Map[Polygon,<<dv-4]}
dv3={EdgeForm[],FaceForm[RGBColor[0.9,0.5,0],
   RGBColor[0.9,0.5,0]],Map[Polygon,<<dv-5]}
dvc={Thickness[0.0008],RGBColor[0.6,0.4,0.5],
   Map[Line,<<dv-4c]}
moh={PointSize[0.005],RGBColor[0.2,0.6,0],Map[Point,<<moho]}
slb={EdgeForm[],FaceForm[RGBColor[0.7,0.9,1],
   RGBColor[0.7,0.9,1]],Map[Polygon,<<slab]}
meq=Map[Point,<<microearthq]
leq={PointSize[0.015],RGBColor[0.3,0.1,0],
   Map[Point,<<largeearthq]}
ref={FaceForm[RGBColor[0.4,0.2,0.5],RGBColor[0.4,0.2,0.5]],
   Map[Polygon,<<reflector]}
vol={PointSize[0.02],RGBColor[0.9,0.2,0.5],
   Map[Point,<<volcano]}
nej={Thickness[0.0065],Map[Line,<<NEJapan]}
Show[Graphics3D[{dv1,dv2,dv3,dvc,moh,slb,meq,leq,ref,
        vol,nej}],
    PlotRange → {{138.5,142.5},{36.5,41.6},{-120,0}},
    BoxRatios → {1,1.5,1.1},
    ViewPoint → {-2.84,-1,1.15},
    Axes    → True,
    Lighting → False]
```

Listing 1. Programs for displaying 3D structure (Figure 1) of the Earth's interior beneath northeastern Japan. Parameter values given in **`ViewPoint`** give Figure 1a. The files, dv-3, dv-4, dv-5, dv-4c, moho, slab, microearthq, largeearthq, reflector, volcano, and NEJapan consist of the data of –3% velocity perturbation, less than –4% velocity perturbation, less than –5% velocity perturbation, –4% velocity perturbation, the Moho discontinuity, the upper plane of the subducting Pacific plate (slab), hypocenters of microearthquakes, hypocenters of large crustal earthquakes, S wave reflectors, locations of active volcanoes, and the coastline of northeastern Japan, respectively.

```
dv1={PointSize[0.001],RGBColor[1,0.3,0.7],Map[Point,<<dv-3]}
dv2={EdgeForm[],FaceForm[RGBColor[1,1,0],RGBColor[1,1,0]],
   Map[Polygon,<<dv-4]}
dv3={EdgeForm[],FaceForm[RGBColor[1,0.74,0],
   RGBColor[1,0.74,0]],Map[Polygon,<<dv-5]}
dvc={Thickness[0.0008],RGBColor[0.6,0.4,0.5],
   Map[Line,<<dv-4c]}
moh={PointSize[0.0065],RGBColor[0.3,0.9,0],
   Map[Point,<<moho]}
slb={EdgeForm[],FaceForm[RGBColor[0.8,1,1],
   RGBColor[0.8,1,1]],Map[Polygon,<<slab]}
meq={PointSize[0.01],RGBColor[0.2,0,0.77],
   Map[Point,<<microearthq]}
leq={PointSize[0.014],RGBColor[0.3,0.3,0.3],
   Map[Point,<<largeearthq]}
ref={FaceForm[RGBColor[0.9,0.63,1],RGBColor[0.9,0.63,1]],
   Map[Polygon,<<reflector]}
vol={PointSize[0.02],RGBColor[1,0,0],Map[Point,<<volcano]}
nej={Thickness[0.0065],Map[Line,<<NEJapan]}
a=0.15; b= -12; c=170
For[i=0,i<c, Show[Graphics3D[{dv1,dv2,dv3,dvc,moh,slb,meq,
             leq,ref,vol,nej}],
      PlotRange → {{138.5,142.5},{36.5,41.6},{-120,0}},
      BoxRatios → {1,1.5,1.1},
      ViewPoint → {-3.6,i*a+b,3.2},
      Lighting → False];i++]
```

Listing 2. Programs for displaying 3D movie (Movie 1) of the Earth's interior beneath northeastern Japan from SSW to NNW.

```
dv1={PointSize[0.001],RGBColor[1,0.3,0.7],Map[Point,<<dv-3]}
dv2={EdgeForm[],FaceForm[RGBColor[1,1,0],RGBColor[1,1,0]],
   Map[Polygon,<<dv-4]}
dv3={EdgeForm[],FaceForm[RGBColor[1,0.74,0],
   RGBColor[1,0.74,0]],Map[Polygon,<<dv-5]}
dvc={Thickness[0.0008],RGBColor[0.6,0.4,0.5],
   Map[Line,<<dv-4c]}
moh={PointSize[0.0065],RGBColor[0.3,0.9,0],
   Map[Point,<<moho]}
slb={EdgeForm[],FaceForm[RGBColor[0.8,1,1],
   RGBColor[0.8,1,1]],Map[Polygon,<<slab]}
meq={PointSize[0.01],RGBColor[0.2,0,0.77],
   Map[Point,<<microearthq]}
leq={PointSize[0.014],RGBColor[0.3,0.3,0.3],
   Map[Point,<<largeearthq]}
ref={FaceForm[RGBColor[0.9,0.63,1],RGBColor[0.9,0.63,1]],
   Map[Polygon,<<reflector]}
vol={PointSize[0.02],RGBColor[1,0,0],Map[Point,<<volcano]}
nej={Thickness[0.0065],Map[Line,<<NEJapan]}
c=157
For[i=0,i<c,  {If[i<17.5,{a=0.3,b= -15}],
      If[And[17.5<i,i<32.5],{a=0.2,b= -13.3}],
      If[And[32.5<i,i<124.5],{a=0.15,b= -11.7}],
      If[And[124.5<i,i<139.5],{a=0.2,b= -17.9}],
      If[139.5<i,{a=0.3,b= -31.8}],
   Show[Graphics3D[{dv1,dv2,dv3,dvc,moh,slb,meq,leq,ref,
         vol,nej}],
     PlotRange → {{138.5,142.5},{36.5,41.6},{-120,0}},
     BoxRatios → {1,1.5,1.1},
     ViewPoint → {-3,0.15,i*a+b},
     Lighting → False]};i++]
```

Listing 3. Programs for displaying 3D movie (Movie 2) of the Earth's interior beneath northeastern Japan, from the sky to the inside of the Earth from the west.

Here we focus our attention on low-velocity zones (where temperatures are potentially high, as described earlier), because they are closely tied to magma source regions and volcanoes. In Figure 1, S wave reflectors (purple rectangles) hypocenters of large crustal earthquakes (brown circles) and low-frequency microearthquakes (black points) are also shown. Although the actual size and shape of the reflectors are smaller and more irregular than those shown in the figures, they are represented by dipping rectangular planes. Large crustal earthquakes of magnitude above 6 ($M > 6$) are based on observations since 1931 [3], and have caused at least some damage in property and human life.

For 3D animations, we successively change the parameters of the function **`ViewPoint`** as described in Listings 2 and 3. The speed, range, and axis of the rotation of the 3D structure are varied by the values, a, b, and c. In Movie 1 (Movie1.nb), y values in **`ViewPoint`** are varied from –12 to 13.35 at every 0.15 (see Listing 2), and there are 170 figures viewing from SSW to NNW. On the

other hand, z values in **ViewPoint** are varied from –15 to 15 at every 0.15, 0.2, and 0.3 in the ranges of $|z| < 6.9$, $6.9 < |z| < 9.9$, and $9.9 < |z| < 15$, respectively (see Listing 3), in Movie 2 (Movie2.nb). This yields 157 figures viewed from the sky to the inside of the Earth from the west. Increasing a value with increasing absolute value of z yields a relatively homogeneous rotation of the 3D structure. For clear presentation of Movies 1 and 2 on the screen or on the monitor, we have used larger parameter values in **RGBColor** in Listings 2 and 3 than in Listing 1, and microearthquakes and large earthquakes are shown in blue and gray circles, respectively. The movies enable us to see the structure from various directions and to investigate the spatial relations between volcanoes, earthquakes, reflectors, and low-velocity zones in detail.

■ Distribution of Volcanoes and Hot Low-Velocity Zones, and Mantle Dynamics

Some interesting features observed in the 3D structure beneath NE Japan are summarized as follows. At 110 km depth, fairly low-velocity regions (velocity perturbations below –4%) appear in the mantle wedge and exist to shallow depths right beneath the Moho discontinuity. Extremely low-velocity regions (velocity perturbations below –5%) are observed around 70 km depth beneath lat. 40.2 N, long. 140 E, and around 40 km depth right beneath the volcanoes. These low-velocity regions appear to be localized. Low velocity indicates high temperature (and so low density) and the possible presence of magma in the region. Diapir-like melting regions of up to a few tens of kilometers are expected along the middle of the mantle wedge. The most extensive melting may occur in the uppermost mantle (~40 km depth) beneath the volcanoes. The continuous supply of magma from melting zones may cause the surface volcanic activity of the island arc. However, most of the magma will cool at depth and form the island arc crust. The local existence of hot low-velocity zones observed in this study may account for the lifetime of the island arc volcanoes. Once a melting zone cools down, volcanic activity will cease. New volcanic activity may start again after another diapir in the mantle wedge rises up to the Moho discontinuity.

The contours of –3% velocity perturbation are also shown in Figure 1 and Movies 1 and 2. Within the contours, velocity perturbations are below –3%, and thus higher temperatures and lower densities than average are expected. These high-temperature regions indicate the upwelling of hot mantle materials from beneath. Two major upwelling regions exist below 60 km depth at 38N and 40N, branch off at the uppermost mantle, and eventually underplate the Moho right beneath the volcanoes. The results indicate that melting zones localized in the upwelling hot mantle materials cause volcanic activity in the NE Japan arc.

In between the volcanoes and the low-velocity zones, mid-crustal S wave reflectors exist, and low-frequency microearthquakes occur in the crust, as also shown in Figure 1. The close relationship of magma source regions with S wave reflectors and microearthquakes indicates that magma ascent from the uppermost mantle to the ductile lower crust may cause low-frequency microearthquakes,

and magma penetration into the brittle upper crust may produce the mid-crustal reflectors, as has been discussed previously [2]. Magma movement in the brittle upper crust may increase stress (or at least change the state of stress) in the crust, and may contribute to the occurrence of crustal earthquakes including destructive earthquakes. Interestingly, large crustal earthquakes ($M > 6$) also occur around the volcanoes and the low-velocity zones (Figure 1, Movies 1 and 2).

■ Summary

The 3D movies are a powerful tool to investigate the internal structure of the Earth. *Mathematica* offers a simple and convenient way to show 3D animations for all sorts of tomographic data. The 3D animations are easily shown on portable computers; therefore, it is possible to present them in lectures, meetings, exhibitions, schools, fieldwork, and so on.

■ Acknowledgments

We are very grateful to Professor I. Kushiro for his helpful suggestions and comments, and Professor D. Zhao for kindly providing the P wave velocity data. This work is supported by grants from JSPS (C12640467) and from the Earthquake Research Institute of the University of Tokyo (2001-A-18 and 2002-A-18). We acknowledge reviewers for thoughtful suggestions and comments.

■ References

[1] M. Mizoue, "Deep Crustal Discontinuity Underlain by Molten Material as Deduced from Reflection Phases on Microearthquake Seismograms," *Bulletin of the Earthquake Research Institute, University of Tokyo*, **55**, 1980 pp. 705–735.

[2] A. Hasegawa, A. Yamamoto, D. Zhao, S. Hori, and S. Horiuchi, "Deep Structure of Arc Volcanoes as Inferred from Seismic Observations," *Philosophical Transactions of the Royal Society of London Series A*, **342**, 1993 pp. 167–178.

[3] A. Hasegawa, D. Zhao, S. Hori, A. Yamamoto, and S. Horiuchi, "Deep Structure of the Northeastern Japan Arc and its Relationship to Seismic and Volcanic Activity," *Nature*, **352**, 1991 pp. 683–689.

[4] D. Zhao, A. Hasegawa, and S. Horiuchi, "Tomographic Imaging of P and S Wave Velocity Structure beneath Northeastern Japan," *Journal of Geophysical Research*, **97**, 1992 pp. 19909–19928.

■ Additional Material

Available at www.mathematica-journal.com.

Movie1.nb. Movie of the 3D Earth's structure beneath northeastern Japan from SSW to NNW. Vertical scale is exaggerated 3.2 times. When running the movie, select the

animation direction of the forward-backward mode (see *The Mathematica Book*: Section 2.10.12). The animation should run from the first cell to the last, and should then reverse back to the first cell again.

Movie2.nb. Movie of the 3D Earth's structure beneath northeastern Japan from the sky to the inside of the Earth from the west. Vertical scale is exaggerated 3.2 times. When running the movie, select the animation direction of the forward-backward mode.

How to Color a Graph with (Computer) Algebra

Yuri Matiyasevich

Steklov Institute of Mathematics at Saint-Petersburg
27 Fontanka, Saint-Petersburg, 191011, Russia
E-mail: yumat@pdmi.ras.ru
URL: http://logic.pdmi.ras.ru/~yumat

This note presents some nonevident ways for determining the colorability of a graph and the number of its colorings via algebraic manipulations of polynomials in many variables.

Suppose we have a graph **G** and are interested in learning whether one can properly color its vertices in **k** colors. In other words, we want to know whether one can assign a *color* **c[i]** to the **i**-th vertex in such a way that the ends of each edge of **G** would get different colors, **c[i]** belonging to some **k**-element set **Colors[k]**. *Could we find the answer with general computer algebra systems*?

On one hand, such systems are universal in the sense that they can compute everything that can be computed in principle, so the answer is definitely positive.

On the other hand, computer algebra systems were initially intended for dealing with mathematical formulas, so the required program might be rather cumbersome. To make our problem more challenging, let us state it in the following way:

Q: *Write the shortest possible program* **ColorableQ** *which would determine, for a given graph G and a natural number k, whether graph G is vertex colorable in k colors*.

Pay attention that we do not ask for the *most efficient* program; in fact, the intrinsic computational complexity of graph coloring is not yet known.

The solution of our problem depends, of course, on the particular system exploited. A specialized system can contain **ColorableQ** as a standard function,

making the solution of our problem trivial. Using the *Mathematica* standard add-on package *Combinatorica*, one can easily define **ColorableQ** as follows.

```
In[1]:= << DiscreteMath`Combinatorica`

In[2]:= ColorableQ[G_, k_] := ChromaticPolynomial[G, k] > 0
```

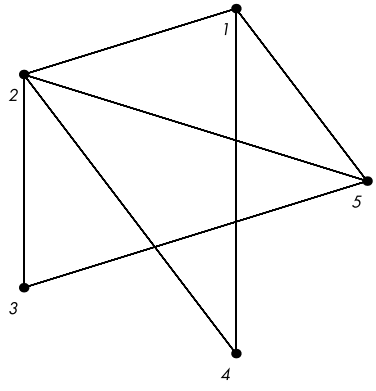
Thus, in order to keep our problem challenging, we need to restrict the admissible tools to basic algebraic manipulations with formulas.

Another point to be taken into account is the representation of graphs. *Combinatorica* uses an adjacency matrix in its canonical representation for graphs. However, it also allows a more compact representation via an unordered set of adjacent vertices. We will be looking for a solution to our problem with this alternative representation.

```
In[3]:= ExampleG = {{1, 2}, {2, 3}, {3, 5}, {1, 4}, {2, 4}, {1, 5}, {2, 5}};

In[4]:= ShowLabeledGraph[FromUnorderedPairs[ExampleG]];
```

From In[4]:=



Here is one possible program. Is it the shortest one based on "first principles" only?

```
In[5]:= ColorableQ[G_, k_] := Module[{c, d, i},
          (Expand[Apply[Times, Apply[Subtract, Map[c, G, {2}], 1]]] /.
            c[i_]^d_. :> c[i]^(d mod k)) =!= 0]

In[6]:= ColorableQ[ExampleG, 2]

Out[6]= False

In[7]:= ColorableQ[ExampleG, 3]

Out[7]= True
```

Let us see why it works. To this end we first rewrite the program in a lengthy form with notation for the intermediate objects.

```
In[8]:= ColorableQ[G_, k_] := (
          Clear[c];
          P1 = Map[c, G, {2}];
          P2 = Apply[Subtract, P1, 1];
          P3 = Apply[Times, P2];
          P4 = Expand[P3];
          DegreeReduction = c[i_]^d_. :> c[i]^(d mod k);
          P5 = P4 /. DegreeReduction;
          P5 =!= 0)
```

The **Map** (applied at level 2) transforms **G**, a list of numbers of adjacent vertices, into a list of pairs of colors which should be different.

```
In[9]:= ColorableQ[ExampleG, 3];
        P1

Out[10]= {{c[1], c[2]}, {c[2], c[3]}, {c[3], c[5]},
          {c[1], c[4]}, {c[2], c[4]}, {c[1], c[5]}, {c[2], c[5]}}
```

The first **Apply** produces a list of differences of colors which should all be different.

```
In[11]:= P2

Out[11]= {c[1] - c[2], c[2] - c[3], c[3] - c[5],
          c[1] - c[4], c[2] - c[4], c[1] - c[5], c[2] - c[5]}
```

As soon as we replace all **c[i]** by some values (from our **k**-element set **Colors[k]**) that do *not* produce a coloring of the graph **G**, the list **P2** will contain a zero.

The second **Apply** transforms **P2** into a polynomial.

```
In[12]:= P3

Out[12]= (c[1] - c[2]) (c[2] - c[3]) (c[1] - c[4])
          (c[2] - c[4]) (c[1] - c[5]) (c[2] - c[5]) (c[3] - c[5])
```

This polynomial is equal to zero for any choice of values of **c[i]** from **Colors[k]** that is not a coloring of the graph.

P4 is another representation for the polynomial **P3**, so it is equal to zero for each choice of values of **c[i]** from the set **Colors[k]** if and only if the graph **G** has no colorings in **k** colors. The size of **P4** is exponentially greater than the size of **P3**, and this is the point that makes our program computationally nonefficient.

The substitution **DegreeReduction** is the heart of the program. In graph theory, colors are not specified. We are free to select for **Colors[k]** any set with **k** elements, and our choice is the set of all **k**-th roots of unity.

```
In[13]:= Colors[k_] := Table[Root[c^k - 1, d], {d, 1, k}]
```

Under this interpretation of colors, the left-hand side of **DegreeReduction** is equal to its right-hand side, so the values of the polynomials **P4** and **P5** coincide as long as the values of **c[i]** are taken from the set **Colors[k]**.

We saw that if the graph **G** has no colorings in **k** colors, then both polynomials **P4** and **P5** are equal to zero for each choice of values of **c[i]** from the set **Colors[k]**. But the polynomial **P5** has degree at most **k-1** in each variable, and by the interpolation theorem there is only one such polynomial. In other words, the graph **G** has no colorings in **k** colors if and only if the polynomial **P5** is *identical* to the zero polynomial, and that is what is checked in the last line of the program.

Thus, there is no magic behind **ColorableQ**. But what is required if we ask about the number of colorings rather than about their mere existence?

Q: *Using only basic tools for formula manipulation, write the shortest possible program* **NumberOfColorings** *which would determine, for a given graph G and a natural number k, the number of vertex colorings of the graph G in k colors.*

It turns out that a slight modification of **ColorableQ** enables us to solve this more difficult problem. First of all, we need to replace the standard **Subtract** by a special function.

```
In[14]:= ColorSubtract[c1_, c2_] := 1 - Sum[c1^d c2^(k-d), {d, 1, k}] / k
```

Similar to **Subtract**, the value of **ColorSubtract[c1, c2]** is equal to 0 as soon as **c1** and **c2** are equal elements from **Colors[k]**. However, if **c1** and **c2** are unequal elements of this set, then the value of **ColorSubtract[c1, c2]** is equal to 1, because in this case the sum is equal to $(c1^k - c2^k)/(c1 - cw)$.

Modifying the final part of the former program, we get the desired new one.

```
In[15]:= NumberOfColorings[G_, k_] := (
           Clear[c];
           P1 = Map[c, G, {2}];
           ColorSubtract[c1_, c2_] := 1 - Sum[c1^d c2^(k-d), {d, 1, k}] / k;
           P2 = Apply[ColorSubtract, P1, 1];
           P3 = Apply[Times, P2];
           P4 = Expand[P3];
           DegreeReduction = c[i_]^d_. :> c[i]^(d mod k);
           P5 = P4 /. DegreeReduction;
           P6 = P5 /. c[i_] -> 0;
           P6 * k^V[FromUnorderedPairs[G]] )
```

P6 is just the free term of **P5**. It turns out to be equal to the number of colorings divided by $k^{V[G]}$, where **V[G]** is the number of vertices of graph **G**. Why?

Due to the fact that the value of **ColorSubtract** is either 1 or 0, the value of **P4** and **P5** is equal to **1** or **0** depending on whether the values of **c[i]** produce a coloring or not. Having a small degree in each variable, the polynomial **P5** can again be restored from these values by the interpolation theorem. The Lagrange form of the interpolating polynomial would consist of exactly **NumberOf**‥**Colorings[G, k]** summands, and it is not difficult to check that the free term of each summand is equal to $\mathbf{k}^{-\mathbf{V[G]}}$.

Removing redundant notation, we get the following short form of **NumberOfCol**‥**orings**.

```
In[16]:= NumberOfColorings[G_, k_] :=
          Module[{c, d, i}, (Expand[Apply[Times, Apply[
                  1 - Sum[#1^d #2^(k-d), {d, 1, k}] / k &, Map[c, G, {2}], 1]]] /.
              c[i_]^d_. :> c[i]^(d mod k) /. c[i_] -> 0) * k^V[FromUnorderedPairs[G]]]

In[17]:= NumberOfColorings[ExampleG, 4]

Out[17]= 96
```

The latter output can be compared with

```
In[18]:= ChromaticPolynomial[FromUnorderedPairs[ExampleG], 4]

Out[18]= 96
```

In the author's HTML paper,
http://logic.pdmi.ras.ru/~yumat/Journal/Triangular/triang.htm, one can find more nonevident algebraic ways for calculating the number of colorings of plane triangulations, giving new restatements of the Four Color Theorem.

About the Author

Yuri Matiyasevich is the head of the Laboratory of Mathematical Logic at Steklov Institute of Mathematics in St. Petersburg, Russia. His most known achievement was the negative solution of Hilbert's tenth problem (http://logic.pdmi.ras.ru/Hilbert10/stat/stat-eng.html) about which he wrote a book (http://logic.pdmi.ras.ru/~yumat/H10Pbook/index.html, English translation published by the MIT Press in 1993). Besides logic, Yuri Matiyasevich works in number theory and discrete mathematics.

Clipping Polygons

Garry Helzer

Department of Mathematics
University of Maryland
College Park, MD 20742
gah@math.umd.edu

A routine for clipping polygons has a variety of graphics applications.

■ Introduction

The standard *Mathematica* function **ParametricPlot3D** requires the parameter domain to be a rectangle. Suggestions for handling nonrectangular domains usually involve triangulation of the domain and a version of the function **TriangularSurfacePlot** from the **ComputationalGeometry** package (see [1] and [5]). The package **SurfaceClip.m** takes a different approach. A rectangle containing the parameter domain is first paved with a set of rectangles, which are then clipped to the shape of the domain. The parametrization functions are then applied to the vertices of the clipped polygons.

As an example, Figure 1 shows a portion of a complicated surface called the bisectrix of the tetrahedron (see [2], [3], [4]). The three pieces of the figure are congruent to the piece of surface shown in Figure 2.

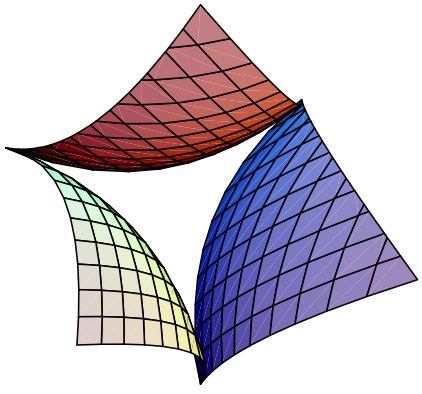


Figure 1. Three surface patches.

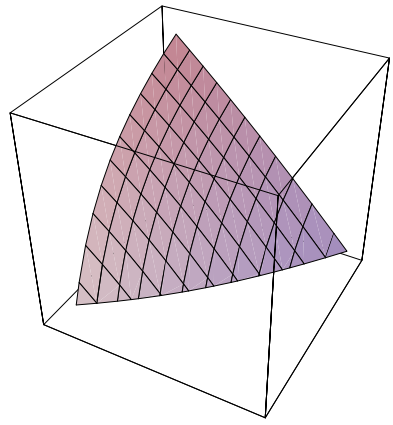


Figure 2. A single patch.

The single patch has a reasonably simple parametrization ($x = -3 + 2\ p + 1/\ p\,q$, $y = 3 - 2\,q + 1/\ p\,q$, $z = -3 + 2\ p + 2\,q$), but to plot the edges correctly the parameters must be restricted to the portion of the unit square in the p-q plane lying above the hyperbola $2\ p\,q = 1$ (Figure 3). To draw Figure 2, a grid of rectangle polygons covering the square was first defined (Figure 3). This polygonal grid was then clipped using the expression $1 - 2\ p\,q$, giving the set of polygons shown in Figure 4. Clipping is accomplished by discarding polygonal vertices lying in the region where $1 - 2\ p\,q$ is positive. Finally, the parametric equations were applied, mapping each remaining polygon to a three-dimensional polygon. (The three pieces of Figure 1 are images of the piece of Figure 2. They were computed using homogeneous transformations from the Wickham-Jones packages [5].)

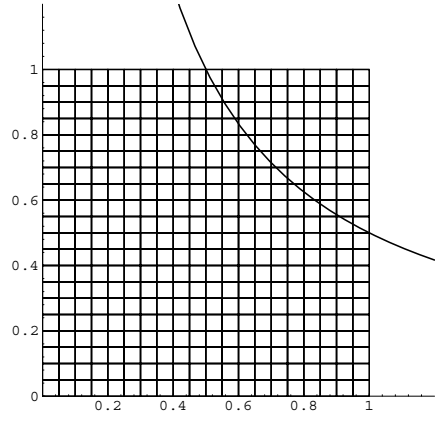


Figure 3. Paving a rectangle.

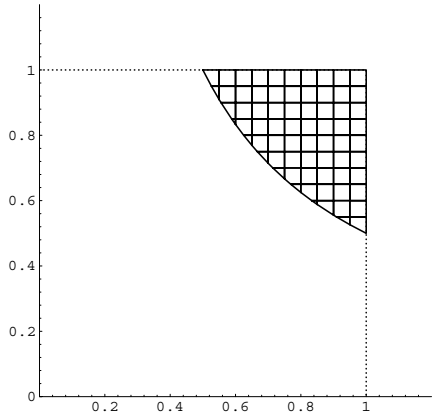


Figure 4. Clipped paving.

The package **`SurfaceClip.m`** contains the function **`ClippedParametricPlot3D`** that automates this procedure. The function takes four arguments and the syntax is that of the built-in function **`ParametricPlot3D`** with an extra argument added. The first three arguments of the two functions are the same, defining the parametrization functions and a rectangle in the parameter plane. The last argument is a list of clipping functions that are applied in succession.

■ Clipping a Polygon

The basic routine of the package is a function that clips a polygon along a curve given by an implicit equation $f(x, y) = 0$. The coordinates of a vertex are substituted into f. Those vertices for which the result is positive are discarded and those for which the result of the substitution is negative or zero are kept. Figures 5 and 6 illustrate the process for a square clipped by the circle $f(x, y) = x^2 + y^2 - 1 = 0$.

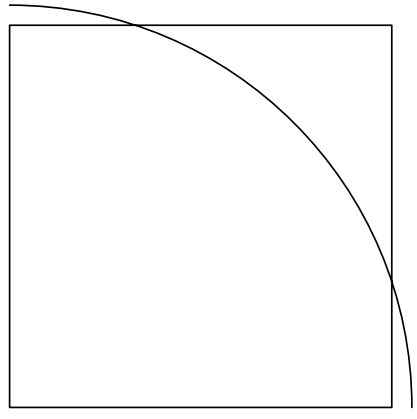


Figure 5. Circle and square.

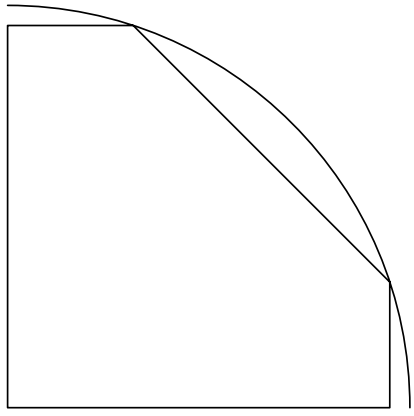


Figure 6. Clipped square.

In Figure 5 a square polygon, shown in outline, is crossed by the circle, only a portion of which is shown. In Figure 6 the polygon has been clipped by discarding the vertex that lies outside the circle and adding vertices where the circle crosses the sides of the polygon. The clipped polygon lies entirely within or on the clipping circle.

The new vertices are computed as follows. Suppose that p and q are adjacent vertices on which the clipping function f has different signs. The edge joining p and q may be parametrized as $l(t) = (1 - t)\ p + t\,q$, and then $g(t) = f(l(t))$ is a real-valued function of the real variable t, which has at least one root between 0 and 1. A root is located with **`FindRoot`** and the value substituted into $l(t)$ to define the new vertex.

Since vertex coordinates are converted to approximate real numbers for this computation, problems with roundoff may occasionally arise. For this reason the test for retaining a vertex is not $g(t) \le 0$ but $g(t) \le \epsilon$ where ϵ is small number internally referred to as **`Fuzz`**. The default value of **`Fuzz`** is 10^{-5} but may be changed by the user.

■ Clipping with Surfaces

Nothing in the above description of the clipping process restricts the polygons to two dimensions or the clipping function to two variables. The process may be used to clip one surface by another. This extends a function in the Wickham-Jones package that clips a surface with a plane. Here are three examples.

Figure 7 shows the Monkey Saddle, $z = x^3 - x\,y^2$, plotted over the square $-1 \le x \le 1$, $-1 \le y \le 1$. The three-fold symmetry of the surface may be emphasized by clipping it to the inside of the cylinder $x^2 + y^2 = 1$. The clipped surface is shown in Figure 8.

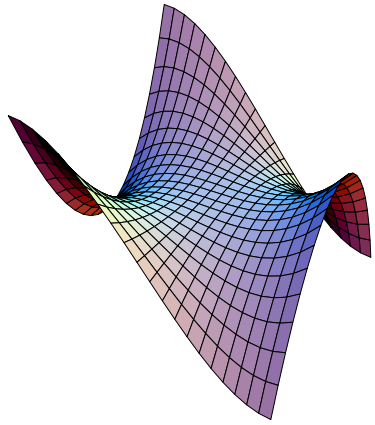


Figure 7. $z = x^3 - x\,y^2$.

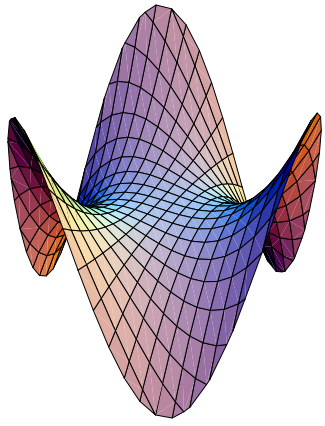


Figure 8. Clipped by $x^2 + y^2 - 1$.

Clipping a plane by a surface gives a section of the surface. An application of this idea is shown in Figures 9 and 10. The Gaussian and mean curvatures of a surface at a point may be defined in terms of the curvature of the curve of intersection of the surface with a plane through the point and containing the surface normal. Figure 9 shows a surface ($z = x\,y$) and a sampling of planes containing the normal to the surface at a fixed point. Clipping the planes by the surface reveals the normal sections. Figure 10 shows the sections with maximal and minimal (signed) curvature. The package contains a utility, **`Grid`**, to generate planes to be clipped.

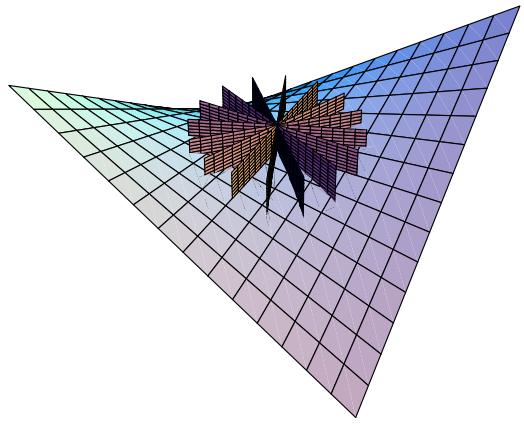


Figure 9. Normal planes.

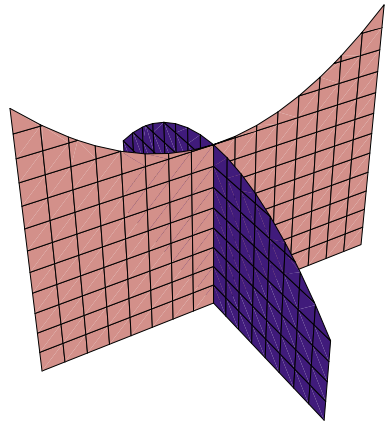


Figure 10. Normal sections.

In Figure 11 the surface is generated by rotating a limaçon about its line of symmetry. A hole has been punched in the surface to reveal the inner lobe. Figure 12 indicates how the clipping was accomplished. The figure shows a cross section (clipped by a plane) of the rotated limaçon together with the surface used to punch the hole. The punch surface is also a surface of revolution. The profile curve is $z = x^8$ rotated about its axis of symmetry—yielding $z = (x^2 + y^2)^4$—and then moved into position by an appropriate geometric transformation.

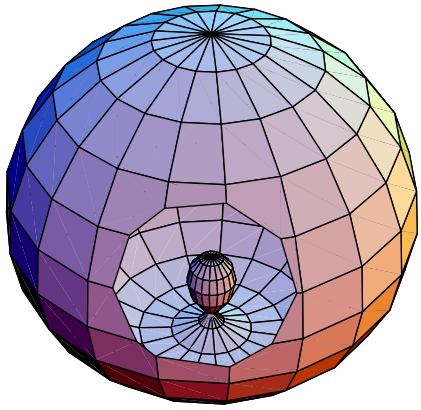


Figure 11. Rotated limaçon.

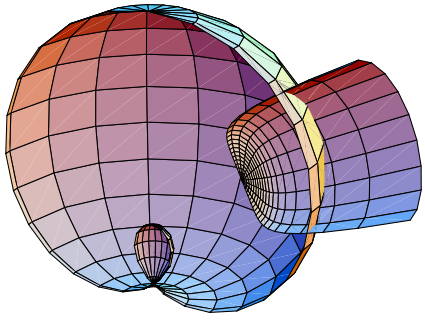


Figure 12. Punch in position.

■ Conclusion

The package **`SurfaceClip.m`** contains the function **`ClipGraphics`** that clips one surface by another. **`ClipGraphics`** takes three arguments. The first argument is a **`Graphics`** object, a **`SurfaceGraphics`** object, a **`Graphics3D`** object, or a list of these. **`SurfaceGraphics`** objects are converted to **`Graphics3D`** objects before clipping. The second argument is a clipping function or list of clipping functions, and the third argument is a variable list. The variable list matches variables with coordinates. The variable list {*p*, *q*, *r*} indicates that the variable *p* refers to the first coordinate, the variable *q* refers to the second, and the variable *r* to the third. In addition to polygons, **`ClipGraphics`** clips lines and points as well.

References

[1] P. Abbott, "Graphs over Convex Polygons," *The Mathematica Journal*, **14**(2), 1994 pp. 21–22.

[2] W. A. Beyer and B. Swartz, "Bisectors of Triangles and Tetrahedra," *The American Mathematical Monthly*, **100**, 1993 pp. 626–640.

[3] G. Gunther and J. B. Wilker, "The Bisectrix of a Tetrahedron," *Mathematika*, **39**, 1992 pp. 93–103.

[4] G. Helzer, "The Bisectrix of a Tetrahedron; An Application of Symbolic Computation," *Mathematica in Education and Research*, **6**(2), 1997 pp. 2–13.

[5] T. Wickham-Jones, *Mathematica Graphics*, New York: Springer–Verlag, 1994.

Article Submissions to The Mathematica Journal

The Mathematica Journal publishes articles and notes on all aspects of *Mathematica* and related subjects. Its goal is to inform and challenge the community of *Mathematica* users and to enable readers at all levels of proficiency to use *Mathematica* more effectively. A high standard of *Mathematica* programming, using clarity and conciseness in writing, is valued. *The Mathematica Journal* reserves the right to edit contributions, with changes subject to the authors' approval.

The Mathematica Journal also publishes *Mathematica* notebooks and packages, as well as programs in other languages that can be used to complement *Mathematica*. All programs are accompanied by a note or article. Programs distributed through the journal can be used freely and duplicated for noncommercial purposes.

Some of the categories of material published in *The Mathematica Journal* are listed below.

- Reports of applications of *Mathematica* to particular fields in education, research, or business. The level of exposition should be such that readers from other fields can grasp the essence of the application, the reasons why *Mathematica* is being used, and the possible relevance to other fields of the techniques being employed.
- Descriptions of programs—*Mathematica* packages and notebooks, or *Mathematica*-related programs in other languages—whether or not accompanied by the code itself. The programs should be exemplary in their implementation and thoroughly tested.
- *Mathematica* programming advice. This can range from notes about little-known programming tricks to tutorials focusing on a general problem or technique. The main test of applicability here is how many readers are likely to benefit from the ideas, but even a relatively esoteric application can aid in the teaching of good programming principles.
- Discussion of issues beyond *Mathematica*. Essays, columns, and surveys on subjects that do not involve *Mathematica* directly may be considered, if they are likely to be of general interest to the *Mathematica* community. Possible subjects include symbolic computation, numerical computation, computer-based education, computer graphics, computer art, and so forth. Articles of this type should be particularly well written and well

argued, and their relevance to the *Mathematica* community should be clear.

- General informational articles, profiles of exemplary *Mathematica* users or applications, interviews, and news related to the *Mathematica* community will also be sought for publication online on a frequent basis.

To submit a contribution, send the material—together with a cover letter stating that it is being submitted for publication in *The Mathematica Journal*—in one of the following formats. Use filenames that suggest their contents.

Web
Articles, graphics, and other files may be uploaded on the Internet at www.mathematica-journal.com/upload.

Email
Articles and other material may be sent by email to editor@mathematica-journal.com. Do not email articles or attachments that exceed 1MB.

Hard Copy and/or Physical Media by Mail
Send to:

Editor
The Mathematica Journal
PO Box 6359
Champaign, IL 61826-6359 USA

Articles, packages, graphics, notebooks, and programs submitted for publication can be sent on cdrom, zip, or other disk. Submissions will not be returned. Do not send proprietary or confidential information.

■ Manuscript Submission Guidelines

Manuscripts must be in English. Manuscripts should be submitted as properly formatted notebooks made with *Mathematica* version 3 or later. Notebooks should be well structured, making good use of sections, text, and inline and displayed equations (if required), and should use the *Mathematica* style sheet "TMJv4Style.nb". This style sheet along with an authoring palette can be downloaded from www.mathematica-journal.com/submissions.

Manuscripts should present a significant application of *Mathematica*. Manuscripts under consideration by or submitted simultaneously to other journals will not be considered.

Remember to include the following:

1. the title, if the title has more than 50 characters, submit an alternative short title;
2. your name(s), affiliation, postal, and electronic mail address;
3. an abstract of at most 150 words.

The remaining material should be included in the following order:

1. text of the article or note;
2. references (see below);
3. a list of additional electronic material, including, if appropriate, information on the necessary environment (machine, operating system, and so on);
4. a biography of each author, limited to 60 words per author, and 180 words overall.

References should be numbered in the text in order of appearance. In the listing at the end of the article, each reference should include full information: author or institution, full title, publisher, city, and year (for books, manuals, etc.); or full journal name, volume, year, and page range (for papers). References to software and hardware should contain complete manufacturer's or distributor's names and addresses. See format in a current issue.

All references in the bibliography should be cited in the text, or accompanied by comments stating their relevance.

Avoid footnotes if possible. If used, they should be numbered sequentially and clearly marked in the text, with the footnotes as a section at the end of the article.

Please observe the following guidelines for text and embedded code.

- Use of the word *Mathematica* should be kept to a minimum, since the reference is almost always clear from the context. It is especially discouraged in titles.
- Technical terms unlikely to be known to nonspecialists should be italicized the first time they are used, and, unless the first use is self-explanatory, they should be defined or explained then. Italics should not be used for emphasis, except very occasionally.
- *Mathematica* words, symbols, or expressions should appear in a font distinct from that used for text. The use of a fixed-pitch font, in which all characters have the same width, is strongly encouraged. Use the author's palette and *The Mathematica Journal* style sheet.
- References to *Mathematica* functions should not be followed by `[]`, unless what is meant is that the function is being invoked with no arguments.
- Lines of displayed code, whether from packages or interactive sessions, should be no longer than 60 characters.
- Do not use *Mathematica* comments in interactive sessions; instead, alternate explanatory text using complete sentences with input/output pairs. It is especially important to explain the commands in a long interactive session as you go along, rather than all at once at the beginning or at the end of the session. As a rule, notebooks should be cast into this same format: explanatory text alternating with code.
- Always mention which version of *Mathematica* you are running.

■ Guidelines for *Mathematica* Packages

Packages are programs written in *Mathematica*, to be used on their own or integrated with the user's code.

The single most important consideration in designing a *Mathematica* package for distribution is that it must be readily usable by others. Clarity is the goal and a good test of the clarity of your design is the ease with which it can be documented. Ideally, the purpose and use of each symbol visible to the user should be simple enough to explain in its usage string.

The more simply your design can be described, the more likely people are to be able to understand and use what you have built. In addition, such designs are often more general and therefore usable in a wider range of circumstances.

The book *Programming in Mathematica*, Third Edition, by Roman Maeder (Addison-Wesley, 1996) provides good pointers on programming style.

Classifieds

Classified advertising is provided for *Mathematica* users. Rates are available at www.mathematica-journal.com/advertising.

***Mathematica* Training and Consulting**—Whether you are an individual or a company seeking *Mathematica* training, Wolfram Education Group provides a learning solution with **Certified Training**. Register for classes and review the Education Benefits Program for corporate account training incentives and discounts. If you are looking for **Independent Training** for additional courses, workshops, or tutorials, the Wolfram Education Group can provide a variety of learning opportunities from independent trainers and companies with years of *Mathematica* experience. Wolfram Research also offers direct **Consulting Services**, such as reviewing and debugging your *Mathematica* code, isolating problems in large *Mathematica* programs, implementing custom features, and writing complete programs based on users' specifications.

More info: www.wolfram.com/services/training

Custom Calculus Solutions Instantly—Automatic step-by-step derivatives and integrals with each step explained. Long multiplication and division of polynomials, too.

More info: www.calc101.com

WHAT OUR CUSTOMERS ARE SAYING:

"One of my indispensable tools."

"Truly a superb numerical software product."

"I have rarely seen such good support from a company."

Global Optimization *for Mathematica*

Global Optimization is a *Mathematica* package for solving constrained and unconstrained nonlinear problems. Accepts black box models, non-differentiable functions, and functions with local minima. On the market since 1998, it has been sold in over 20 countries. Received enthusiastic review in Oct. 2000 Scientific Computing World.

- FIVE GENERAL CONSTRAINED NONLINEAR SOLVERS
- INCLUDES INTERVAL METHOD SOLVER
- CONSTRAINED NONLINEAR REGRESSIONS AND MLE
- TABU SEARCH & INTERCHANGE METHOD FOR 0-1 INTEGER PROBLEMS
- EASY-TO-READ MANUAL
- FREE UPGRADES FOR 1 YEAR

For more information or to buy online, visit:

www.wolfram.com/products/applications/globalopt

Developed and supported by Loehle Enterprises
1258 Windemere Ave. Naperville, IL 60564 630.527.0785 craigloehl@aol.com