Mathematical Visualization and Online Experiments

Konrad Polthier

Technische Universität Berlin
polthier@math.tu-berlin.de

Abstract. The future of mathematical communication is strongly related with the internet. On a number of examples, the present paper gives a futuristic outlook how mathematical visualization imbedded in the internet will provide new insight into complex phenomena, influence the international cooperation of researchers, and allow to create online hyperbooks combining interactive experiments and mathematical texts.
Using the software JavaView we discuss practical aspects of online publications and give technical details on the ease of implementations. The online version of this paper is a sample interactive document with visualization examples and numerical experiments.

1 Introduction

Visualization has a major impact on the understanding and exploration of complex mathematical phenomena. As in other sciences, images are used in mathematics as helpful illustrations accompanying textual descriptions, as a communication medium to exchange ideas with other cooperating researchers, and to explain deep mathematical results in a comprehensive way to non-experts, just to mention only a few applications.

The process of visualization is a synonym for a broader process than the production of images or video animations. Prior the generation of images, abstract mathematical concepts must be translated into discrete descriptions which are numerical data structures to hold the mathematical information. For example, a smooth surface must be discretized in a set of triangles, and algorithms must be translated from their smooth counterparts acting on smooth geometries to discrete geometries.

In general these conversions are delicate tasks. For example a Riemann surface may have different smooth descriptions requiring non-obvious discrete structures such as triangulated piecewise linear surfaces with specific discrete properties. Continuous Riemann surfaces are identified under conformal maps, and so one would like to perform a similar identification on the corresponding discrete objects. It is of central importance in mathematical

[0] to appear in: M. Emmer (Ed.), Matematica e Cultura 3, Springer Verlag 2000

visualization to have discrete equivalents of smooth concepts, and discrete algorithms on discrete surfaces which perform the same operation as their counterparts on smooth concepts.

The process of defining useful discrete data structures, finding good discretization algorithms, and deriving methods operating on discrete geometries belongs to the same category of mathematical tasks which have been so important for smooth geometries. Good visualization and numerics rely on perfect discrete definitions which are the basis to assign, measure and transform mathematical properties of discrete geometries. Numerical mathematics has gone a long way in discretizing function spaces but, for example, the understanding of the intrinsic mathematical properties of discrete geometric shapes still leads to challenging questions.

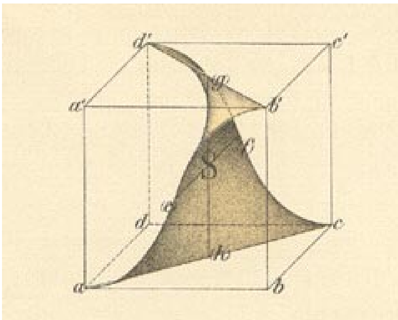

Fig. 1. Copper plate engraving of the Gergonne surface by Hermann Amandus Schwarz. Image taken from [1].

It is important to distinguish between the process of visualizing precomputed numerical data and, in other sciences, of experimentally measured data, and the process of doing numerical experiments and simulations. The first process is an analytic process trying to understand large sets of numbers by the means of finding good visual representations. This includes hiding and emphasizing of data as well as the feature extraction of detail information. Many visualization tools operate in a post-processing step by evaluating large data sets and generating new smaller sets with feature information. For example, one of the current major problems is the visualization of turbulent 3d flows. Because of its complexity the flow might be unaccessible to a direct visual representation but the path of moving vortices may be computed and easier to display. Here the correct definition of vortices in a discrete flow is an essential prerequisite before starting the development of a tracing algorithm. The second process of doing numerical experiments and simulations is a constructive and repetitive process where an experiment

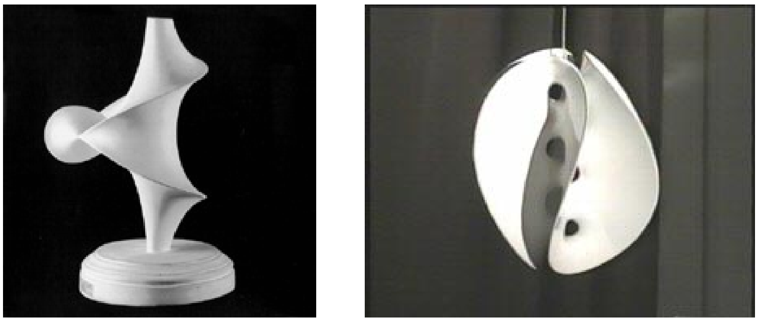

Fig. 2. Plaster model of the Kuen surface. (courtesy Gerd Fischer) (left). Stereo lithography model of Chen-Gackstatter-Karcher-Thayer surface. (right).

is analysized during runtime and information obtained to steer parameters and interact with the numerical process. Here visualization is used to obtain insight into the behavior of a running algorithm rather than into geometric dataset.

Over the past years, visualization has proven to be a successful tool for mathematicians in the investigation of difficult mathematical problems which seem to be unaccessible by standard mathematical tools. It has proven its potential by substantially contributing to the solution of hard mathematical problems.

Mathematicians often have a concrete imagination of abstracts shapes even in higher dimensional spaces, and especially geometers have a deeply visually related thinking. Images are very helpful during their research, they give insight into unknown phenomena and suggest directions for further investigations. Nevertheless there is a remarkable contrast between the visual thinking, and the occurrence of pictures in research publications and inner-mathematical talks. Often images are extremely rare, for example, books on differential geometry, a mathematical subject with one of the largest fundus of easily visualizable complex geometric shapes, contain only a handful figures with most simplest shapes. The lectures on differential geometry of Luigi Bianchi is a comprehensive and very influential book on surface theory, whose German translation was published in 1910, and the classic book on Riemannian Geometry by Gromoll, Klingenberg, and Meyer from 1969, which was the book for a generation of geometers, both books contain no single image of a surface or a geometric shape. Similar examples up to modern days could be listed.

In some sense images have the character of an experiment: they suggest a truth or result but usually do not have the power of a proof. An image

Fig. 3. Deformable thread model of a hyperbolic paraboloid (left). Apollo Belvedere with parabolic curves (right). (Courtesy Gerd Fischer)

can illustrate a mathematical proof but an experiment or image implicitly requests a formal proof of its suggested result. Nevertheless, important experimental results exist which have not been rigorously proven, but they direct theoretical investigations and often give final hints. Experiments have been performed in the whole history of mathematics but only since 1991 there exists a publication media, the journal of Experimental Mathematics.

In contrast to a wide-spread opinion, mathematical history contains a rich set of visual examples. Archimedes was drawing figures into the sand when being bothered by the Romans during the capturing of Syracuse. Euclidean geometry, nowadays referred to as 'elementary geometry', is one of prominent examples where all kinds of drawings always played a central role in the communication and publication of results. Famous non-trivial examples are the copper plate engravings of Hermann Amandus Schwarz shown in figure 1 [10]. He discovered new minimal surfaces which solved long-standing questions in geometry and analysis on the existence of solutions to elliptic boundary value problems. Although his proof was of theoretical nature he found it worthwhile to invest a lot of energy to include visual images of the new surfaces into his research publications. This is remarkable if compared to the much less effort one needs for the generation of computer images today.

One of the most thorough approaches in mathematics to use physical models and experimental instruments in education and research is the famous collection of mathematical models in Göttingen. This model collection already had a long history when Hermann Amandus Schwarz and Felix Klein overtook the direction of the collection. Especially under the direction of Klein the collection was systematically modernized and completed for the education in geometry and geodesy. This collection was considered so important that Klein exhibited the models on the occasion of the World's Columbian Exposition 1893 in Chicago [2]. The models were produced among others by the publisher Martin Schilling in Halle a.S., see his catalog of mathematical models [9]. The price of approximately $250 per model was relatively, and therefore, the large size of the collection of more than 500 plaster models is even more impressive. The collection can still be seen in the mathematical department in Göttingen, and a description including photos of many models is given in Fischer [4]. An example is the plaster model of the Kuen surface in figure 2 which is shown together with a modern model of the Chen-Gackstatter-Karcher-Thayer minimal surface produced in stereo lithography technique from digital data. The production of models slowed down and finally stopped in the beginning 30s. Following Fischer, the reasons were not purely economical nature but also the appearance of more general and abstract view points in mathematics. It was the time when the books of van der Waerden with only simple illustrations, and of Nicolas Bourbaki with a complete ignorance of images appeared.

It seems that minimal surfaces have been among those geometric shapes which often urged mathematicians and physicists to produce images. In the sequel to Schwarz and the experiments of Joseph Antoine Ferdinand Plateau, it was Richard Courant, Johannes C.C. Nitsche, and Alan H. Schoen who experimented with physical soap films and even produced real models for permanent display and for easier communication. Among the first breakthroughs of mathematical visualization was the proof of embeddedness of another minimal surface. Celsoe Costa discovered the mathematical formulae of a genus 1 minimal surface which was a candidate to solve a 200-year long question: whether there exists a third embedded and complete minimal surfaces with finite total curvature beside the trivial examples, the flat plane and the rotational symmetric catenoid. David Hoffman and William Meeks developed computer programs to visualize Costa's surfaces and watch surface properties which they could later successfully prove after having enough insight into the complex shape of the surface, see figure 4 and [7].

2 Experimental Online Geometry

It has been pointed out by David Epstein and Silvio Levy [3] that "the English word *prove* – as its Old French and Latin ancestors – has two basic meanings: to try or test, and to establish beyond doubt. The first meaning is largely archaic, though it survives in technical expressions (printers proofs) and adages (the exception proves the rule, the proof of the pudding). That

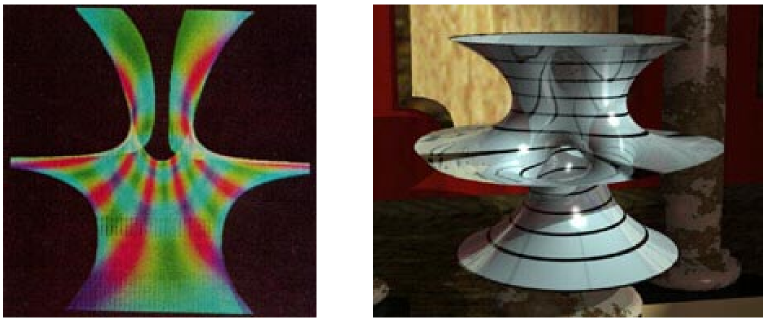

Fig. 4. Unveiling the Costa formulas: first image of the surface over not appropriate parameter domain and frame buffer artifacts (courtesy Jim and David Hoffman) (left). Years later, raytraced image ot the Costa-Hoffman-Meeks minimal surface. (right).

these two meanings could have coexisted for so long may seem strange to us mathematicians today, accustomed as we are to thinking of proof as an unambiguous term. But it is in fact quite natural, because the most common way to establish something in everyday life is to examine it, test it, probe it, experiment with it."

Computers are nowadays a powerful machinery to perform mathematical experiments, and even do automatic proving as soon as it is possible to formulate axioms and rules in a formal computer language. A large part of applied mathematics is devoted to the simulation of practical physical phenomena which often are still inaccessible to formal proves of, say, convergence.

3 Geometry on the Internet

For a number of years the internet has been a technical infrastructure connecting computers to a global network, but in recent years it has emerged as the world-wide-web, a global information network similar to a gigantic hypertext book. Hypertext books have been around for quite some years, for example, digital help systems of software applications have been among the first to use software links between different sets of informations. Mainly the success of hypertext books made written manuals obsolete so that nowadays software is solely accompanied by digital manuals. Classical hypertext books have a similar linked structure as the web but the information is usually stored locally. The new dimension of the web allows to imagine the human's knowledge as a worldwide digital encyclopedia which is directly accessible by all people rather than by those living in the vicinity of the local library.

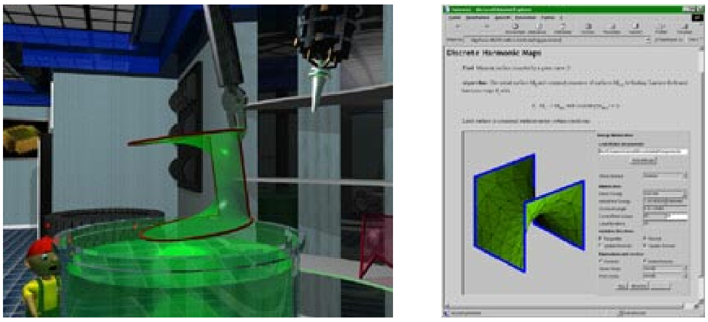

Fig. 5. Soap film machine from video *Touching Soap Films* [1] (left). Computing minimal surfaces in an interactive applet on the JavaView web site [8] (right).

The success of the world-wide web was not planned. It is one of the achievements that happen in the modern computer world where suddenly a far-reaching answer to the problems of a large number of people appears. Among the major reasons of the success of the world-wide-web are the global standards which have been established for the document format HTML, the network protocol HTTP, and the software browsers. While the network protocol is based on the internet protocol TCP/IP which has been used in the academic domains for a number of years, it was the simple new document format HTML which helped to attract so many people. This format is easy to read and to write on nearly all computer systems, and it allows to include multimedia elements like images, sound, video, and other data. From the beginning of world-wide-web, browsers had a simple user interface which allowed immediate access to all kinds of information and thereby succeeded to replace complicated software tools previously used in the academic internet.

The origin of the world-wide-web is a good example of another aspect of the web beside the publication and presentation of information, namely, communication. In fact, the HTTP protocol, which is the underlying technical protocol for internet connections, was defined at CERN in Switzerland to support communication and exchange among large groups of scientists involved in physical experiments. Often these groups involve more than hundreds of scientists located at different places in the world who must exchange experimental data and communicate research results in efficient ways. This means that the origin of the web is not related with presentation, which made the web popular, but with communication among people. The communicative aspect of the web is becoming again its original importance and is the main attraction of the web nowadays.

When considering the web as a global digital library, is there still a place for maintained libraries and encyclopedias? In fact, such managed collections are still important since they ensure a certain quality, a managed database, and well-defined access. The global web contains a huge amount of information but it is often hard, and becomes even harder, to find the good information among the non-relevant. Information must be preselected to be really useful, and current search engines are just starting with methods for ranking and sorting information by quality.

Geometry is the content in this setting which we are now going to exhibit in greater detail. The special characteristic of geometry is its rich fundus of shapes, images, and dynamic applications which make it an ideal candidate for profiting from the internet.

One of the most attractive new components of the web is the ability to present and communicate complex interactive experiments. Performing experiments has always been the domain of experts, more concrete, of those experts who wrote the simulation software. One reason is the user interface design which has been difficult to create for unix computers since user interface builders are not so common in this area. Another reason is the lack of incentive for a programmer to create a well-thought user interface since scientific programs usually require specialized hardware and will seldom reach a wide-spread audience. Since the arrival of the new programming language Java in 1995 the situation changed dramatically. For the first time it is possible to create software which runs on any computer platform and operating system without the delicate process of rewriting and adjusting source code for each new system.

Java is usually installed automatically on a computer at the same time when a browser is installed. In contrast to other high-level programming languages like Fortran, C or C++, the language Java comes also with a full-featured set of graphical user interface structures. These structures are not supplied by other parties but incorporated into the definition of the language itself. The design of a user interface is one of the major tasks when implementing a reusable experiment, and in previous times it was the one of the reasons why software was restricted to special computer platforms. Platform dependent interface code restricts software to distinguished platforms. Since Java includes the graphics user interface directly in its language specifications, Java programs run per default on any computer with installed Java.

4 Visualization and Online Experiments

Mathematical visualization has proven to be an efficient tool for analyzing complex mathematical phenomena, and it has given decisive hints leading to rigorous mathematical proofs of long-standing problems. Visualization is

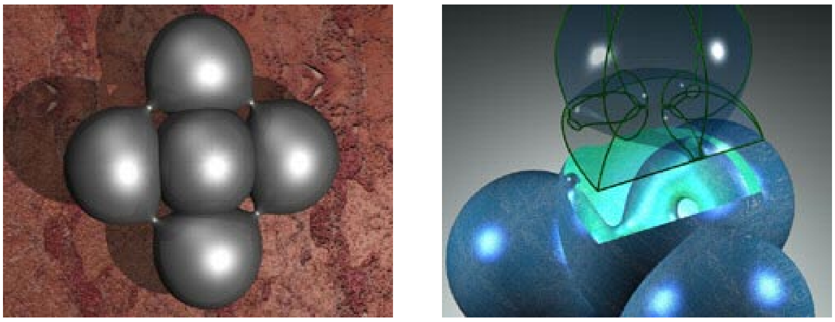

Fig. 6. Compact soap bubble with genus four (left) and tetrahedral symmetry (right).

not only a tool to visualize complex objects but in combination with modern numerical methods allows to perform mathematical experiments and simulations in an artificially clean environment. For example, the unveiling of the Costa-Hoffman-Meeks surface [6] (figure 4), or the first numerical examples of compact constant mean curvature surfaces with genus greater than two [5] (figure 6) are among the most prominent results of the fruitful interaction of mathematics with the new toolkit mathematical visualization.

Mathematical experiments require the following goals to be accepted in the mathematical community similar as experiments in physics and chemistry:

1. Validation of experimental data by independent groups
2. Publication and storage of experimental results and data sets
3. Cooperation of researchers on the same experiment while sitting at different places

Up to now visualization has required high-end workstations combined with mainframe computers for numerical computations. Each research lab had developed own software dedicated to specific graphics hardware. The specialization of the software for a specific visualization task or numerical problem as well as the dependence on a specific hardware platform had major drawbacks for the scientific communication. The major reason for the current non-fulfillment of the goals 1. and 2. is a missing worldwide standard and interface for the different software packages, and for goal 3. the missing standard to exchange data sets between experimental software packages and research publications. Currently research publications are paper-based where there is no way back from written publications to digital data sets. Even worse, in order to fit into a restricted page layout and fulfilling the allowed number of pages, publications cannot include all data but restrict to the so-called 'most relevant data'. The presentation of experimental results in publications are incomplete and often moved to the appendix so that it is usually not possible to validate or reproduce them.

In order to allow validation of experiments and of numerical data sets it is essential to allow direct public access to the data in an electronic format in the same way as public access to research publications is given through libraries. This joint publication of experimental and numerical data requires to insist on the similar principles known from the publication of scientific research results:

1. Unambiguous, self-contained digital representation of the data, i.e. no dependence on existing software packages.
2. Reviewing of data sets, e.g. to ensure technical correctness and scientific relevance.
3. Indexing data sets to allow unique references, e.g. when data is used in other experiments.

In the following sections we discuss how online visualization might look like and about the possible difficulties to encounter.

4.1 Workstation versus Online Visualization

For a long time scientific visualization was beyond the budget of many mathematical departments. Large research institutes, military organizations, and commercial companies were among the first who could afford specialized graphics hardware. In science, specially funded research groups where able to afford high-end graphics workstations including the necessary staff to manage the machines and simultaneously do the scientific experiment. In the mean-time, the computational power of personal computers with relatively cheap graphics card suffice to perform most of the scientific visualization tasks found in research. Nevertheless, one still encounters the following drawbacks of the current software running on specialized workstations and mainframes:

1. Specialized and expensive graphics hardware.
2. Large program size since operating system just supports basic functionality.
3. Usually only the programmer is able to run the experiments.
4. Installation at other sites requires experts, and does not allow regular update.
5. Advantage: extremely fast execution speed.

These drawbacks are in strong contrast to the situation we have encountered during the development and usage of the software JavaView. JavaView is a scientific visualization software completely written in the programming language Java. Java is an object-oriented programming language similar to the language C and C++ but different in the sense that Java is designed to run on any computer. Further, Java programs may run inside web browsers. Both properties are the reason that Java has become the major programming language for interactive web applications since its first presentation in 1995. A program written in Java has the following advantages:

1. Runs on Standard PC and Workstation.
2. Tiny program size because Java base classes are already installed.
3. Each application has a user interface per default since it runs in a browser.
4. No installation beside a browser with Java since browser performs the data transfer.
5. Speed: depends.

These advantages have the following reasons:

(1.) Java is automatically installed on a computer if a web browser is installed. Therefore, the popularity of web browsers helped to install Java on nearly any computer world-wide.

(2.) The size of Java programs is usually very small compared to classical stand-alone application software since the Java base classes, which are comparable to software libraries, are already installed. Therefore, an application must only deliver its additional functionality, and not system routines.

(3.) An application inside a web page must have a well-designed graphical user interface since it is by default used by some other people than the programmer. This is in contrast to classical experimentation software, and leads to a great benefit in the design of better products.

(4.) The installation of classical software systems has often been a pain. The customer often needed to compile the package again on his machine, or make special adjustments depending on his specialized hardware. The author was in an even worse situation. He needed to offer and maintain different versions for different platforms. When using Java then there exists only one version independent of the hardware platform and operating system. This is possible since the Java virtual machine must cope with system differences, so the responsibility is transferred from the author of applications to the supplier of the Java virtual machine. Therefore, the installation process of a Java application such as JavaView is reduced to downloading an archive, i.e. one or more library files, which is done automatically through a web browser. This allows the author to concentrate on the development of the software without keeping to much care on the destination platform, and it frees him from providing installation mechanisms. The user is freed from any installation task, he just starts his browser and selects a Java enhance web page.

(5.) The speed of Java applications not only depends on the hardware but to a large extend on the quality of the installed Java virtual machine (JVM). A Java application consists of machine independent byte code which is interpreted by a JVM and executed on a local computer. JVMs differ largely in quality, for example, when loading a Java application some JVMs compile the byte code into machine dependent code, which leads to a drastic increase in execution speed.

5 JavaView

JavaView [8] is a software for sophisticated experiments and visualization of 2- and 3-dimensional geometric objects on a local computer as well as online in a web browser. Students, teachers and researcher can use JavaView as a tool for general scientific visualization, in a distant learning environment, for online exchange of scientific results among researchers, and for electronic publication of mathematical experiments.

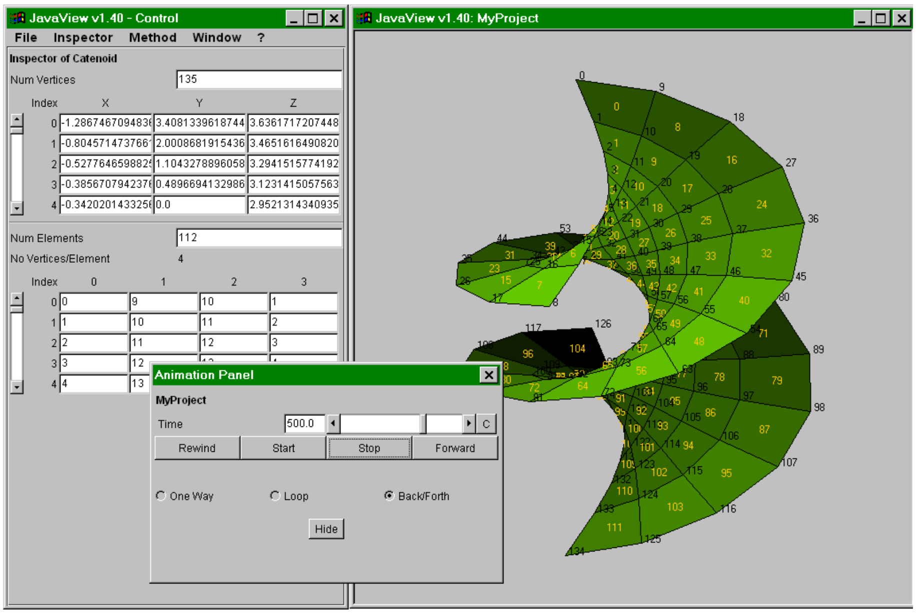

Fig. 7. Interactive exploration of geometries and animated sequences of transformations.

JavaView is a numerical software library with a 3D geometry viewer written in Java. It allows to add interactive 3D geometries to any HTML document, and to present numerical experiments online. The future of mathematical communication is strongly related to the internet, and JavaView enhances classical textual descriptions not only with images and videos but additionally with interactive geometries and online experiments.

JavaView has been developed to solve the following technical tasks:

1. Visualization of mathematical data sets inside web pages.
2. Interactive experiments and simulations inside web pages.
3. Inclusion of mathematical experiments and simulation data in electronic publications.

The first version of JavaView fulfilling these tasks was released in November 1999 after development versions had been used in geometric research projects at the Technische Universität Berlin for over a year. JavaView is now used at different places world-wide. There exist a multitude of mathematical demonstrations to give an outlook of the range of new applications possible with web-based experimental software.

The first two tasks provide the technical basis for the third task. As a proof of usability, JavaView has been selected by the project "Dissertation Online" of the German science foundation DFG to produce a reference online dissertation in mathematics. There are many other issues to solve for electronic research publications, and JavaView is only one component that is connected with the inclusion of experiments and interactive visualization. For example, mathematical journals currently require a paper based version of an article even if they include the article in an electronic version of the journal too. An author must create two versions of an article, usually a TeX version for paper-based printing and an online version including, say, JavaView experiments. But even electronic versions of journals are currently not well-suited for web-based usage since they are just PostScript or PDF documents. Such documents do not fit with other internet technologies since, for example, they are hardly searchable and do not allow inclusion of Java applets or video elements.

In the following subsections we discuss some properties of JavaView and give a number of sample experiments. Since this book is paper-based we refer to an online version of this article at the web site http://www-sfb288.math.tu-berlin.de/vgp/ which contains interactive versions of the presented applets.

5.1 Properties

The viewer of JavaView supports most interaction features of an advanced 3D viewer. For example,

- Rotation, translation, zoom, camera control, picking
- Inspection of geometries and material properties
- Selective display of vertices, edges, faces, vector fields
- Animations, keyframes, auto-rotation
- Subdivision and simplification of meshes, adaptive and hierarchical triangulations
- Advanced visualization algorithms, LIC rendering, textured surfaces
- Import and export of geometries in multiple data formats
- PostScript and image file export for inclusion into paper publications
- Frontend for other applications, for example, to view *Mathematica* graphics

Images and geometries exported with JavaView may easily be included in TeX publications as well as online documents. Additionally, JavaView is a class library for advanced numerics in differential geometry including tools for finite element numerics. A standalone version of JavaView runs in a Unix or Windows shell from the command prompt, and can be attached as 3D viewer to other programs like *Mathematica*.

5.2 Online Example Applets

This paper-based publication must restrict to a few sample applications and verbally describe the possible interaction. The following examples describe different aspects of the usage of JavaView:

1. Visualization and evaluation of precomputed models which are stored somewhere on the internet.
2. Interactive tutorials explaining simpler numeric or geometric facts to accompany classical lectures or online learnshops.
3. Sophisticated numerical research projects with intimate combination of numerics and advanced visualization.
4. Joint research of authors at different universities doing numerical research experiments embedded into web pages.

Geometric Models Online Nowadays it is even hard to obtain datasets of simple examples. One reason is the absence of published numerical data sets stored at some publishing house, or a data set which accompanies a research article published in a scientific journal. This is a serious drawback in the scientific validation of numerical experiments but also leads to the more elementary drawback that models are not available, for example, as initial data sets to perform own experiments. Often one would just like to have a digital model with certain properties to test one's own numerical algorithm or implementation. Compare applet 7 for an online animation which may be interactively investigated.

Another aspect of an online model collection is the educational benefit. Geometry books usually contain only a few images, and students still have a hard time to find good visual material. The software JavaView provides a simple way for every scientist to have mathematical models at his fingertips.

Interactive Tutorial: Root Finder Interactive tutorials may be easily designed to explain simpler facts from numerics or geometry. They may be made available to students online as the supplemental material of classical university and high school courses. Additional to other course notes the students may regularly look at web pages accompanying the current course. Distant learning projects must have well prepared course materials since the direct contact with students is less intensive. These projects will be among the first to include interactive experiments, and maybe even the driving forces for the development of whole packages of interactive online experiments.

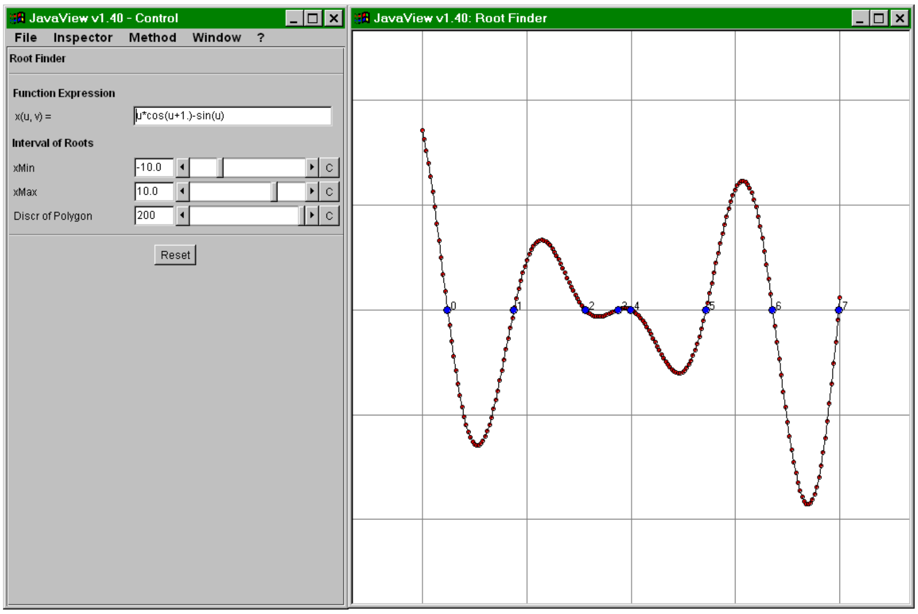

Fig. 8. Applet to find roots of explicitly given functions. Function expression maybe typed and the roots immediately searched.

The applet 8 demonstrates the numerical algorithm of finding the zero-crossings of functions. The method subdivides the original interval and uses

Brent's method to find the roots on each subinterval. Here, the user is assumed to read a description of the algorithm and simultaneously study the online example.

Numerical Experiments and Visualization Java has the same properties than other programming languages used in numerical mathematics and visualization. It has some structural speed limitations since it is an interpreted language but modern just-in-time compiler are able to provide a compilation process on the fly while loading the program. Java is currently not a language of choice for high-performance computing but maybe used in a large section of nearly all numerical areas. The great benefit of Java is its machine and operating system independence, and since most of the time used in numerical research projects is spent in program development and code maintenance it is a great tool even in these domains.

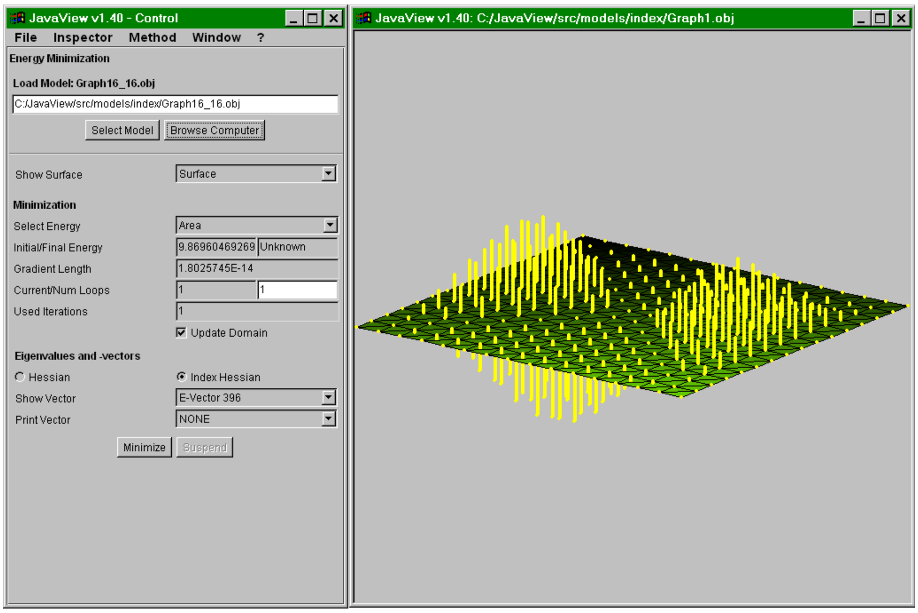

Fig. 9. Study of eigenvalues and eigenfunctions of the Laplace-Beltrami operator on surfaces. Here the known *cos* and *sin* functions are reproduced on a planar square. Other applets show the eigenfunctions of the second variation of area.

Research applications like the eigenvalue computations in figure 9 demonstrate that numerical computations and scientific visualization are possible with JavaView. Further, the coherent interface of Java applets immediately make these experiments accessible by other people, and their inclusion in digital publications.

Case Studies: Research Cooperation Online The cooperation of researchers who are located at different places usually lacks a certain amount of communication. For example, joint numerical experiments require the exchange of newly developed software and their local installation. Here we encounter one of the major benefits of Java: first, the platform and operating system independence, and second, the invisible installation of Java software via automatic web-based mechanisms.

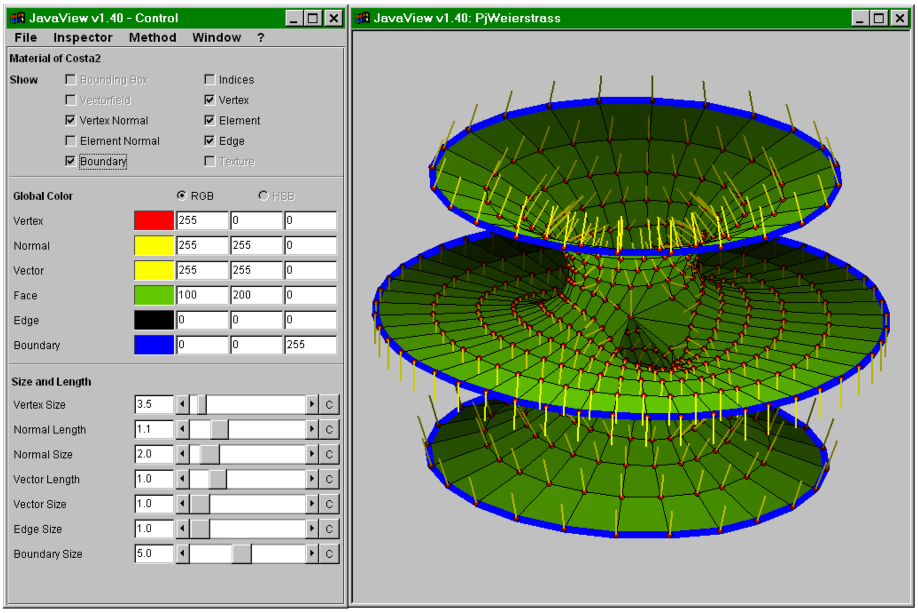

Fig. 10. Animation of the associate family of the minimal surface of Costa-Hoffman-Meeks showing the isometry of the transformation and parallelism of normal vector.

Index of Minimal Surfaces The index computations of the second variation of area of unstable minimal and constant mean curvature surfaces was a joint project with Wayne Rossman in Koebe in Japan. The cooperation started during his visit in Berlin, and was continued via putting our experimental applets online to be accessible world-wide. The software development was continued in Berlin and the software archive on our web site was regularly updated with the newest version. Each time Rossman did an experiment using the web based software on our site he was sure to use the latest software without ever installing any local version.

Social Behavior of Honey Bees In the meantime, JavaView is used in other projects too. For example, biologists in South Africa currently use one of our applets to compute shortest curves in a beehive to get information on the distance of the queen to selected other bees. Here the biologists load a model of the beehive into the geodesic applet, interactively select the positions of two bees, and invoke our geodesics algorithm to obtain the shortest curve connecting both positions. This example stresses the online usage of JavaView as well as the automatic installation process.

5.3 A Hands-On Example

This hands-on example describes the necessary three steps to create a JavaView enhanced web page which displays a geometry model in a small window allowing interactive modifications.

1. Insert a Java applet tag into a web document *myPage.html* referring to a geometry model *brezel.obj*.
2. Get the archive *javaview.jar* from the JavaView homepage.
3. Upload all three files to a web server.

The sample applet tag somewhere inside the document *myPage.html* looks as follows:

```
<applet
     code=javaview.class
     archive=''javaview.jar''
     width=200 height=200>
     <param name=model value=''brezel.obj''>
</applet>
```

This applet visualizes the geometry model inside a small window of 200*200 pixels on the web page. The model need not be a geometry file on a local computer but the model parameter may be any internet address referring to a model on an arbitrary web server.

This example stresses the fact that the installation of the JavaView software is no longer an issue compared to the installation process of other software. The browser keeps care to download the required Java archive when it encounters the archive parameter inside the applet tag. The browser also ensures that the archive is downloaded only during first usage, and later reuses the version it has stored in the browser cache.

The easy download mechanism is especially useful for library servers offering Java enhanced electronic publications. The digital article and the JavaView archive are both stored, for example, in the same directory on the library server. The files must be uploaded by the author as described above, and are automatically downloaded by a browser when a user accesses the web page. Therefore, the librarian has no additional duties related with software installation. The library must only offer the usual upload mechanism for documents which it has already installed.

6 Summary and Outlook

The internet will dramatically change the classical way of communicating and publishing mathematics. We have given some ideas on possible changes to expect, and the benefits which mathematics may gain from these new developments. The interactive, exploratory component of mathematics, which has been removed from mathematical publications for a too long time, is now available in the form of Java enabled software. We have given several examples of multimedia enhanced experiments which allow to imagine the possibilities waiting at the horizon.

For more information and interactive versions of the experiments described in this paper we refer to the JavaView home page [8]. These pages also include tutorial material how to include interactive geometries into own web pages.

References

1. A. Arnez, K. Polthier, M. Steffens, and C. Teitzel. *Touching Soap Films.* Springer VideoMATH Series, 1999.
2. Chicago Historical Society. http://www.chicagohistory.net/history/expo.html.
3. D. Epstein and S. Levy. Experimentation and proof in mathematics. *Notices of the AMS*, June 1995.
4. G. Fischer. *Mathematische Modelle/Mathematical Models.* Vieweg, Braunschweig, 1986.
5. K. Große-Brauckmann and K. Polthier. Compact constant mean curvature surfaces with low genus. *Experimental Mathematics*, 6(1):13–32, 1997.
6. D. Hoffman. Computer-aided discovery of new embedded minimal surfaces. *Mathematical Intelligencer*, 9(3), 1987.
7. D. Hoffman and W. Meeks. Complete embedded minimal surfaces of finite total curvature. *Bull. AMS*, pages 134–136, 1985.
8. K. Polthier, S. Khadem-Al-Charieh, E. Preuß, and U. Reitebuch. http://www-sfb288.math.tu-berlin.de/vgp/javaview/, 1999.
9. M. Schilling. *Catalog Mathematischer Modelle Für Den Höheren Mathematischen Unterricht.* Leipzig, 1911.
10. H. A. Schwarz. *Gesammelte Mathematische Abhandlungen.* Springer, Berlin, 1890.