EG-Models - A New Journal for Digital Geometry Models

Michael Joswig and Konrad Polthier

Technische Universität Berlin, Institut für Mathematik

Abstract. The archive Electronic Geometry Models is a new electronic journal for the publication of digital geometry models from a broad range of mathematical topics. The geometry models are distinguished constructions, counter examples, or results from elaborate computer experiments. Each submitted model has a self-contained textual description and is peer reviewed, and later reviewed by the Zentralblatt für Mathematik.

This paper gives in depth information about the principle ideas behind this service and discusses various technical issues. In particular, we show how XML related techniques are applied.

1 Introduction

We have set up a new journal for electronic geometry models at the website `http://www.eg-models.de/`. The collection exhibits peer-refereed data sets of geometry models from a broad range of mathematical subjects including, but not restricted to, differential geometry, discrete geometry, computational geometry, topology, and numerical mathematics. The published models are intended to provide insight into geometric shapes, to serve as a unique reference for scientific datasets, and to enable the validation of numerical experiments. We do have images and we do have interactive means for visualization. But, it is essential that the focus of this server is not restricted to the visualization aspects. The key item is the data set itself combined with a self-contained description which states why the model is mathematically important [11] [10].

Mathematical models have a long history even if they have been in the background for a longer period. Famous mathematicians like Felix Klein and Hermann Amandus Schwarz started to build up large collections of plaster models in the 19th century for educational purposes and, moreover, mathematical research, see Figure 2. The idea of this electronic model server is to continue the plaster collections with modern computer tools. But the possibilities of the digital models go well beyond those of the libraries with classical plaster shapes and dynamic steel models in earlier days. For one thing the digital models are accessible through web browsers and accessible world-wide. Very many of the models are displayed in a fully interactive viewer. Users can adjust the camera and viewing options, for instance, to simulate a walk through a model.

Much more rewarding, however, is the option to use the data sets on the EG-Models server for one's own computations and experiments.

Many interesting geometric objects are generated by computer programs today, the reason being that they are too large or too complicated to construct by hand. And the Internet is full of interesting collections, but often the data made available is in some obscure format or it lacks information, which makes it hard to use. For our models we have a set of non-proprietary ASCII based standard file formats, which are documented on the server. Many software tools can use these formats right away, and it should even be easy to make one's own software read any of them. Besides this, nonetheless important, technical detail, there is a much more severe problem: the integrity of the data provided. This is why it is crucial to run such a server for digital models like a refereed scientific journal.

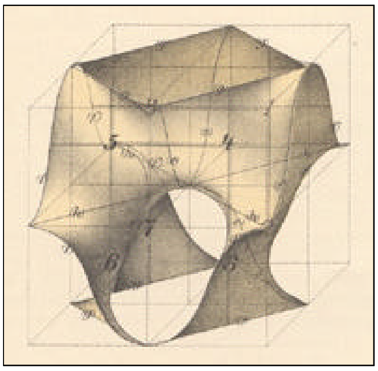
The collected models are from two different categories. First, the archive is open for electronic counterparts of classical known geometries for educational purposes and for reference. Second and most importantly, the server hosts new models and previously unknown results of mathematical experiments whose existence contributes to mathematical knowledge. Models of the second kind are peer reviewed before publication like an article contribution for a mathematical journal. After acceptance and publication these models will further be reviewed by Zentralblatt für Mathematik.

The access to the model server is free of charge. The data of the model server is harvested by and made accessible through qualified search engines such as MathNet's Sigma Searchable Index.

There is a wide range of possible applications for our kind of electronic models: education and analysis of shapes, comparability and verification of experiments, as well as models which further research can be founded on.

The educational aspect was one of the driving forces behind the historic plaster collections. This continues to be true for electronic models. Users may interactively study complex shapes similar to the study of physical geometries. Even more with a digital model, users may interactively study complex shapes much as they study physical geometries. The set of tools operating on normalized shapes will be continuously growing.

Computational experiments based on numerical algorithms are often hard to compare due to individual details in the modeling. In this context it is a major advantage to have a unique model to base further computations upon. Several different research groups may solve the same problem with different algorithms or software packages. Their respective performance can be compared on standard models. Such a comparison of mathematical experiments is currently not possible since classic journals usually do not allow publishing experiments which include complete digital data yet. At most, a description of the experiment contains a few sample figures, which are often useless for other experimenters. One reason is the lack of space in printed journals. But, even with the appearance of electronic journals the classical style of publication of experiments did



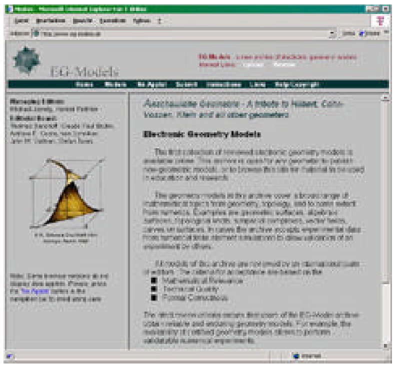


Fig. 1. Copper plate engraving from 1865 by H.A. Schwarz [18] (left). Homepage of the new EG-Models server (right).

not change by much. The EG-Models server not only provides a place for the publication of experiments but it also specifies formal criteria for the data sets to be published. As a further benefit standardized models establish perfect test cases for new algorithms and implementations.

Experimental activities in mathematics often require an enormous setup of software tools and of modeling of the initial data sets. For example, a numerical flow experiment requires the discretization of a physical body. Similarly, the eigenvalue computation of a functional on a geometric surface requires a discrete surface representation. In both cases, the EG-Models archive saves a lot of initial work for the experimenter.

The server grew out of discussions at the conference on the "Future of Mathematical Communication" (FMC) in December 1999 at MSRI, see `http://www.msri.org/calendar/workshops/9900/`. At that time, the software tool JavaView [14] was already developed for doing interactive geometry on the Web and had proven its usefulness in a sample application for the "Dissertation Online" project of the major German academic libraries. Several participants of the FMC conference suggested using JavaView to setup an electronic model server for certified and reviewed geometry on the web. During the beginning of the year 2000, we sketched an outline of the server functionality as well as of the software tools required. To establish specifications of the underlying document data formats turned out to be the most crucial step. All documents for the textual model description as well as the model data format are now specified in the new XML language, see Section 5. The specifications were widely discussed with a number of people. In particular, we are indebted to Bernd Wegner of Zentralblatt and Wolfram Sperber of the MathNet Metadata group for sharing their profound experience on digital publications with us. Starting in summer 2000 we ran the server in a non-public beta test. This included model submissions from various external sites, performing the reviewing process, and testing the software for automatic upload, formal validation etc.

This article presents the basic concepts and working principles of the EG-Models server in detail including technical background information on various data formats and tools for managing the server. We start with a review of physical models in mathematical history in Section 2. Section 3 contains a discussion of the working principles of this model server. This is followed by a description of the submission and refereeing process in Section 4. Background information on data formats and technical issues are given in section 5. Finally, we provide several examples of published models to demonstrate the benefits of this server for mathematical research and education.

2 History of Geometry Models

Mathematical history contains a rich set of visual images and physical models to be used in mathematical experiments or education. 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 the prominent examples where all kinds of drawings played a central role in the communication and publication of results. Famous non-trivial examples are the copper plate engravings of Hermann Amandus Schwarz, see Figure 1. 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 for the production of the copper plate engravings of the new surfaces to be included in his research publications [18].

One of the most thorough approaches in mathematics using physical models and geometric 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 took over the direction and systematically modernized and completed the collection for education in geometry and geodesy. A large number of models were produced by the publisher Martin Schilling in Halle a.S. [17]. The price of approximately $250 per model was relatively high, and therefore, the large size of the collection of more than 500 plaster models is even more impressive.

It seems that minimal surfaces have been among those geometric shapes which often urged mathematicians and physicists to produce and use images. Among the first breakthroughs of mathematical visualization was the proof of embeddedness based on visualization of a minimal surface. Celsoe Costa discovered the mathematical formulae of a genus 1 minimal surface which was a candidate to solve a 200-year old question: whether there exists a third embedded and complete minimal surface with finite total curvature beside the trivial examples, the flat plane and the rotationally symmetric catenoid. David Hoffman and William

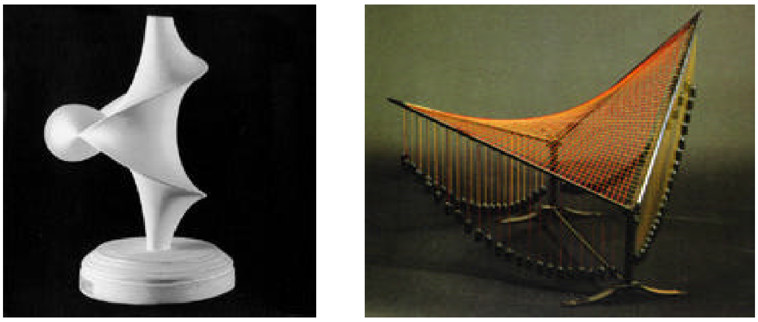


Fig. 2. Plaster model of the Kuen surface (left) and a flexible steel model of a hyperbolic paraboloid (right). Courtesy Gerd Fischer [3].

Meeks [8] 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.

3 Publishing Models in an Electronic Journal

There are at least three important reasons for the publication of refereed geometry models:

1. Validation of experimental data sets
2. New research based on refereed geometry models
3. Establishing intellectual property rights

The publication of mathematical experiments is nowadays widely accepted as a source of inspiration and a test bed for theoretical ideas. Mathematical experiments seem to get an important role similar to experiments in physics. There already exist mathematical journals with a special focus on the publication of experimentally obtained results. For example, *Experimental Mathematics* (`http://www.expmath.org`) and electronic journals like *The Electronic Journal of Combinatorics* (`http://www.combinatorics.org`), *Geometry and Topology* (`http://www.maths.warwick.ac.uk/gt/gtp.html`) or *Documenta Mathematica* (`http://www.mathematik.uni-bielefeld.de/documenta/`) already have a first class status. The dramatic influence on mathematical knowledge acquisistion may be best seen on the effect of preprint servers like the server in Los Alamos (`http://xxx.lanl.gov`) and its multiple mirror sites. Nevertheless, publications on experiments still remain textual descriptions which hardly allow other researchers to validate the experiments using their own software tools, or to use the experimental data as a basis for own experiments.

For example, the Penta surface [6] belongs to a group of compact constant mean curvature surfaces which were experimentally computed with a new algorithm based on discrete constant mean curvature surfaces. The original publication does not contain any experimental data beside the verbal description, and, although in this case the data is available from the authors, we believe nobody is currently able to reproduce the experiment without an enormous, year-long effort.

To illustrate a possible application of certified geometry models, we propose two tasks related with the Penta surface and its companions:

1. Validation of the experiment: These compact CMC surfaces are unstable with respect to direct variational methods, and have not been constructed elsewhere neither numerically nor theoretically. Validate the performed experiments!
2. Usage of experimental data: The degree of instability is measured by the index of the surface. Calculate the index of this compact CMC surfaces based on the data sets underlying the original research publication!

Currently, both tasks cannot be performed within the possibilities of classical journals in mathematics. This is where the new EG-Models journal has its place:

- Development of quality criteria of geometric data sets.
- Certification of geometry models by peer-review.
- Archiving of certified geometry models.

3.1 A Model Submission

A model is represented by a master file which is a unique and well-defined digital data set of the geometric shape or of the experimental result. This master file is the reference data set of the geometry model, it is the central component of the publication. A model may be generated using an arbitrary software tool, but it must be stored in one of a few geometry file formats. A valid submission of a model consists of the following set of files:

- A data file of the model, the so-called master model. Often this file consists of a polyhedral mesh of the shape of the geometry including further information.
- A description file of the model with a self-contained explanation of the model, and author, content and bibliography.
- Optionally, an unrestricted set of additional files for various purposes. For example, a preview image, a Postscript image for inclusion in paper based articles, files for explanatory purposes, or files usable in other software packages.

Table 1. File types in a submission

Purpose	Description	Format
Description	Complete description of model submission. Review decision is based on master file and this description.	xml
Master	The major geometry data file which is the reference data set. All other geometry files have minor importance and are usually derived from this unique file.	jvx, obj, poly, byu
Applet	Small sized version of the geometry for online display in an interactive applet. If the corresponding Master file is small you may reference the Master file as Applet file.	jvx, obj, byu
Preview	Small image for fast preview of the model. Recommended image size is 240×240 pixels.	gif, jpg
Readme	Additional instructions for editors and administrators of data base. This file will not be included in the public presentation of the model. Instructions to readers of EG-Models should be placed in the model decription.	txt
Print	Image of model for inclusion in publications. Note, users of this file must correctly cite the model as being published in this data base. PostScript files must be ASCII without preview image. TIF files must have IBM byte-encoding and LZW compression (loss-less).	ps, eps, tif
Original	If master file has been generated by a software package from this original source, then this original file may also be supplied although there is no guarantee that the EG-Models archive supplies any tools operating on this file.	
Other	Other supplemental files, there is no guarantee that the EG-Models archive supplies any tools operating on this files.	

Description and master are required. Applet and preview are recommended. All other are optional. If no format is specified, then the format is arbitrary.

Table 2. Files in the submission of the model [1]

Retract.xml	The description of the model including author information
Retract_Master.jvx	The master data set of the model
Retract_Applet.jvx	A small version for fast preview in an interactive applet
Retract_Preview.gif	An image for fast preview on the web site.
Retract_Print.ps	PostScript image for inclusion in TeX publications
Retract.fe	Private file format for software Surface Evolver

Table 1 lists the different file types in a submission. For example, the submission of the deformation retract soap film [1] by Ken Brakke consists of the set of files displayed in Table 2.

These files must be produced by an author with his or her private tools or with any software product. The files are then uploaded by the author using the submission page of the EG-Models archive. The upload initiates an automatic verification of the uploaded description and geometry files.

3.2 Guidelines for Good Model Datasets

The master model specifies the shape of the geometry and is the most important data set of each submission. It is important for comparability of models and the verification of experiments that the shape of each model is uniquely defined by its data file. For example, piecewise-linear representations of a curve or surface are usually individual data sets. Algorithmic representations of a model often specify a data set up to some accuracy only since each realization of the algorithm may terminate at a different accuracy.

Most of the popular data formats for geometry exchange fail with at least one of the basic requirements for a long term archival, an automatic syntax validation or a unique reconstruction of the simplicial shape. For example, assume a triangle mesh of a two-manifold with self-intersections where is may easily happen that four triangles have a common edge along the intersection line. Here a data format must describe a neighborhood relation between pairs of adjacent triangles to allow the reconstruction of the two-manifold along the common edge on the intersection line.

The quality of a model depends on the syntax of the data file and on the semantics of the mesh of the model. The specification of the data file must allow a verification of the content of a given data file against a language specification of the syntax and it must allow an unambiguous reconstruction of the model. We have chosen an XML syntax for the data file which allows an automatic validation of a given data file against the corresponding DTD. The JVX format is an XML based geometry format

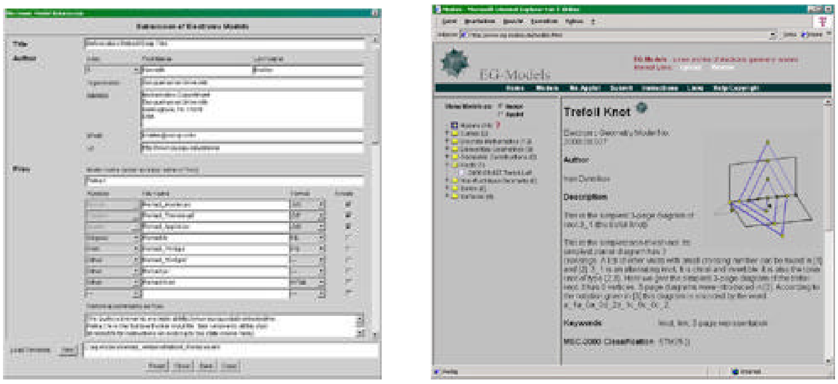


Fig. 3. Online submission form (left) and a published model (right).

for the description of complex geometries and the automatic validation of data sets. It was developed within the JavaView project, and is used as one of the major submission formats.

The formal language specification of shapes has another very important aspect. Normalization allows third parties to invent tools which will operate on any model which follows the same language specification. Not only does this allow exchanging models but also making use of other tools.

3.3 Description of a Model

It is essential that a model submission includes a self-contained detailed description file explaining the mathematical relevance of the model and often details of the experimental setup. The format of a description file must follow exact rules to enable mechanical processing, for example, an automatic generation of a web presentation or print version.

We decided to base most of the EG-Models documents on the XML data format. XML is designed for the automatic validation and the mechanical processing of documents. In particular, XML documents are easily converted into different display formats like, for example, an HTML web page.

Note that an author neither needs to know about XML nor to type any XML by hand. There are different ways to generate model description files, either:

– fill out a template file using the online submission form on the EG-Models server.
– fill out a submission from within JavaView version 2.00 or later, e.g. import your geometry model and let JavaView automatically generate a master file, an applet file, a preview image, and the description file by pressing the “Submit” button in the “File” menu.

- for multiple submissions it might be easier to use a special tool jvSub which is available at the EG-Models site.
- download and complete an XML template file with an ASCII editor (requires basic XML knowledge).

A real-world example of a completed description file is given in the Appendix of this article. As a note for experts, the EG-Models XML description file is formatted according to the data type dictionary available at `http://www.eg-models.de/rsrc/eg-model.dtd` on the EG-Models homepage.

Part of the reason why the plaster and string models from the nineteenth century ceased to be effective as teaching and learning tools is that the accompanying material was usually not displayed, and often it was lost, so that most viewers had no idea what the model stood for. Fischer's [3] two-volume set "Mathematische Modelle" helped to rectify that situation, and updated things at the same time. It seems that we want the new electronic model collection to be at least as useful as those volumes, and that means that we have to provide access to background and supporting materials. This does not have to add a good deal of material to the first description people see, but it does suggest that there can be a reference to a more complete treatment of the model. Since we have a list of models, we should also have a list of additional pages that go with some of the models, and perhaps in some of these pages, we can even give the information someone would need in order to make a model, concrete or electronic, static or dynamic.

At first sight, the descriptions attached to models in the EG-Models archive seem to do better than the loose descriptions in the old plaster collections since access to each model is through its description page. But, if somebody downloads a model from the EG-Models collection then the close link to the accompanying description is lost in some situations, currently. Therefore, we decided to use the possibility of the JVX format to add author and abstract information to a JVX file, too. Once again, the author and abstract information will not only be stored in the accompanying XML description file but also directly inside the JVX model data file, at least the master file.

3.4 Extended Thoughts

Deviating from the descriptive style of the previous sections, here we try to explain the reasoning behind some of the design decisions made. In particular, we focus on some of the restrictions of the current setup.

The appearance of explicit coordinates in a data set seems to look static and opposite to the idea of interactivity. At first sight, an algorithmic representation or at least a functional representation of a data set might look more appealing, especially when thinking about interactive experiments for educational purposes. There are a different views on this issue:

1. The listing of explicit coordinates allows reconstructing the shape of a model with relatively few tools, and very small geometries may even be reconstructed by hand without any software tool. For example, the most common geometry file formats are the very simplest formats since it is very easy to write and read such files even with inhouse software. Simplicity of a file format simplifies the exchange of models among different people and software packages. This is often more relevant than a complete description of all digital properties of a model since often at least subsets of missing properties are not relevant in a special situation.

2. A functional description of a model is unique in the mathematical sense. Nevertheless, algorithms often operate on a discrete mesh to be obtained by a discretization process. This process may introduce new numerical errors depending on the discretization process such that a comparison of results becomes ambiguous.

3. Instead of a static model we might even investigate the possibility to store an interactive experiment, say, a Java applet. And often it is extremely easy to produce such a Java applet. In fact, the whole JavaView project was started to allow easy preparation and inclusion of mathematical experiments in online browsers, and of course, we would be happy to offer this functionality for models in the EG-Models server too. For example, adding the energy minimizer applet to a minimal surface, adding the Platonic solid generator to a Platonic surface, etc. However, the essential question is, how to make a software experiment enduring? With our high-aiming goals for the model server we must think of future times when Java applets simply will not be state-of-the-art any longer or, even worse, are no longer supported by standard browsers. If only 10% of all models come with individual applets, even applets that we wrote ourselves, migrating to a different architecture will be impossible without loosing too much of the functionality.

A partial solution to the problems is a suitable extension to the JVX file format where functional expressions are allowed as coordinates instead of numbers. Often this is more or less a suitable solution, but it is easy to think of models where the situation becomes much more involved: for example, if the choice of several parameters is not independent. To cover the full complexity of possible functional expressions, we will end up in defining a suitable formal language, similar to the programming languages defined by packages like Maple and Mathematica.

We have to be very careful about including things in the JVX format that will be difficult to update, and it is a good service we are providing by raising and examining these questions in a critical way. Since nobody is sure what the general solutions will be, in the meantime, it can be helpful to look at particular examples, as we are doing here. It seems that with regards to a polyhedral surface it would often be easy to include a list of the coordinates of the vertices so that any person who wishes to can enter them into the appropriate program for display, or for that matter make a cardboard model! In contrast, the parameterized version can be turned into an interactive demonstration by anyone with access to an appropriate program. Moreover this flexibility encourages people to think about the important situations where the nature of the model changes, rather than simply observing some fixed example.

If we rely on the data set of a model as the central topic of interest then we are completely independent of the temporal development of technical tools. This data set has the power of becoming a durable part of the mathematical knowledge in the same way as we see it with articles published in mathematical journals. We are stressing this fact so much because we think that durability is one of the most important reasons of the EG-Models project among, of course, quality and others.

We discussed the possibility of including some interactive demonstrations, and there was some concern that not everyone would be able to work with them, and that they might not be of "archival" quality. Still it seems that there is a great demand for allowing interactive experiments on the EG-Models server. As a partial solution, we have decided to accept interactive experiments as an unsupported part of a submission. For example, an author may include additional files (file_other) which provide an experimental setup such as in the retract model in section 8. But note, in spite of their beauty in the long term these files might not be supported by the EG-Models archive.

4 Refereeing Process

Each submission is peer-refereed by at least one editor or an external referee. Based on this review, the formal correctness, mathematical relevance, and technical quality the editorial team decides about acceptance of a model. Such strict review criteria ensure that users of the EG-Model archive obtain reliable and persistent geometry models.

4.1 Formal Correctness

Each model submission must consist of a master file containing the geometric data of the model itself as well as a description, see Section 3. All the files comprising a submission are strictly required to satisfy a variety of formal criteria. This is to allow the file handling on the server done automatically — at least to a large extent.

The master geometry file must fit the format specifications of any of the geometry file formats JVX, OBJ, POLY, BYU. The server contains a full description of these formats at `http://www.eg-models.de/formats/`. These standard file formats are easy to convert into other file formats.

The description file must be an XML file which validates against the data type dictionary (DTD) at `http://www.eg-models.de/rsrc/eg-model.dtd`. This DTD is the formal specification of the XML description file, see Section 5. It is recommended to supply a (GIF) image for previewing and a suitably small (JVX) data set for interactive visualization on the Web. Additional files for information purposes are welcome.

4.2 Mathematical Relevance

Each model must stand in relationship to a distinguished problem of mathematical importance. Typical examples are the following.

- The model yields a counter example to a previously open conjecture.
- The model provides the first numerical solution of a problem.
- The model is another solution to an already solved problem with distinguished new properties.

4.3 Technical Quality

We want to exhibit models which are as useful as possible. Therefore, we require the models to satisfy a range of criteria in addition to what can be formally specified. The individual criteria vary with the area the model comes from. We give examples.

One large class of models on the EG-Models server consists of surfaces (and also manifolds of higher dimension). In most cases the models of these surfaces are, in fact, discrete. It is standard to have a coordinate description for the vertices, augmented by the *mesh*. This mesh carries the information which vertices form the polygonal faces of the surfaces. Depending on the purpose intended, faces may be considered flat or not. In the latter case curvature information is typically encoded as normal vectors at the vertices, which can be interpolated. The preferred file format for surfaces is JVX.

Often for visualization purposes the exact properties of the mesh are of a lesser significance. But, for topological investigations it is crucial to have a description which is correct in the sense that the mesh does encode a triangulation or cell decomposition of the surface. This way the mesh itself already carries the topological information. Exceptions at singularities are unavoidable, of course. The following criteria serve as a guideline:

- Use double precision for floating point coordinates and prefer integers where appropriate.
- Make a triangulation as regular as possible and avoid degenerate numerical situations like thin triangles or triangles with zero area.
- Use a simplicial or polygonal grid without cracks in the mesh and multiple vertices.
- Supply neighbour relation between adjacent triangles or polygons to allow the unambiguous reconstruction of the manifold structure of the mesh.
- Reduce the number of vertices and elements of a geometry model to a reasonable magnitude.
- Provide a uniform orientation of all faces where possible.
- Provide separate geometry models inside a file if the models have different functionality. For instance, for the model of a surface inside a sphere, describe both the surface and the sphere as separate objects. Note that the JVX file format allows encoding several geometrical objects within a single file.

There exist several software tools for optimizing geometries. For example, JavaView allows simplifying meshes of geometric surfaces under curvature control. The technical quality of numerically obtained geometry models is hard to measure, and the editors will decide case by case.

A different class of geometric objects on the server are convex polytopes. The boundary of a d-dimensional polytope is known to be a $(d-1)$-dimensional sphere. So, in principal, whatever has been said about (hyper-)surfaces also applies to polytopes. And, indeed, there are several polytopes on the server which are encoded as JVX. Nonetheles, for most polytopes polymake's [4] genuine file format is preferred. The reason is the following. For an individual polytope model often the *exact* vertex (and facet) coordinates do matter. polymake's file format allows the specification of unlimited precision rational coordinates.

This suffices for most cases, but not for all. Most of the 3-dimensional regular solids, for instance, do not admit a *regular* representation with rational coordinates. The way out is twofold, and, in fact, both options are used for polytope models on the server. The *combinatorial type* of any 3-dimensional polytope can be realized with rational coordinates: so one can describe a 3-polytope with the same face lattice instead and with rational coordinates which do not differ by much from the algebraic numbers which define the corresponding vertices of the regular sibling.

There are even more serious cases, however, which cannot be remedied by the procedure sketched above. It is known that there exist convex polytopes whose combinatorial type cannot be realized with rational coordinates, see Ziegler [20].

5 XML for Descriptions and Models

XML is the successor of HTML and is becoming the new lingua franca of the internet. The XML language plays a role at several stages on the model server. Most importantly, it is used for the textual model descriptions as well as for the geometry files (if the JVX format is used). Internally, it is also used within our refereeing system and for other "housekeeping" purposes. The two main advantages of XML are:

1. Straightforward convertability of the data into various other (including future) formats.
2. Syntax check of the data (validation).

This first property allows displaying the content of the server in different views. For example, a referee of a model sees more than the user of the server. XML provides means to produce views of one model in all kinds of abstractions. Further, the XML format allows an almost automatic conversion of the model data into other formats. HTML, as used on the server, is only one example. The JVX files can easily be converted in standard formats used by other programs such as BYU, OBJ, etc. For a project like the EG-model server it is essential to take into account that none of the currently used file formats will survive forever. XML has the capability to cope with exactly this problem.

“The Extensible Markup Language (XML) is the universal format for structured documents and data on the Web.” This is quoted from the W3C’s web site `www.w3.org`. There is a large set of buzz words combined with prententious statements floating around all areas of the computer related business. XML recently received a lot of attention also by a wider audience. The reason is that XML is designed to be the successor of the Hypertext Markup Language (HTML) which drives our Web. However, it is important to realize that the capabilities of XML go far beyond what is possible to express with HTML. In particular, it is not at all restricted of storing Web content. In the following we will try to give an idea about XML and related technologies.

There are two key properties of XML which are perfectly expressed in its name. Firstly, XML is a Markup Language, that is, XML imposes a structure on the data stored. Here the structure essentially is a rooted tree with arbitrarily many children for each node. This tree structure is the key for processing the data in a standardized way. In contrast to HTML, where the structuring *tags* are mainly used for specifying layout datails, the XML document itself typically stays completely neutral with respect to layout. The second crucial thing about XML is its extensibility. Again a comparison to HTML might help. While the latter consists of a fixed set of tags (which grows and changes from time to time), the former allows almost arbitrary tags; the only restrictions are somewhat similar to restrictions concerning the names of variables for common programming languages. In addition to these tags XML has only a little more to offer. At first sight it might be hard for the experienced programmer to grasp the benefits of this ultimate form of freedom.

5.1 XML Examples from the EG-Model Server

Let us turn to a simplified example of how a model is respresented on the EG-Models Server, see Figure 4. Each model, in fact, comes with a whole bunch of files. But, there is an XML file at the core of it. Much like a journal article it contains a description of the model, references to other sources of information, AMS classification, and the author’s address. Additionally, there is a unique identifier which provides a uniform resource location (URL), that can be used to retrieve the model description from the Web by means of any ordinary browser. Moreover, this XML file contains references to the other files comprising the submission of the author. Among these there is the *master file* of the model, which in this case is again a file in an XML format, namely a JVX file, which is JavaView’s native file format. The example is almost self explaining: The tags start with “`<`” and end with “`>`”. Each tag `<tag>` has a unique closing tag `</tag>`. This is an important difference from HTML where the closing tags are usually optional. Opening and closing tags must be properly nested in order to form a correct XML expression.

```
<eg-model>
  <eg-id>2001.01.042</eg-id>
  <title>Deformation Retract Soap Film</title>
  <modelname>Retract</modelname>
  <authors>
    <author>
      <firstname>Kenneth</firstname>
      <lastname>Brakke</lastname>
      <affiliation> ... </affiliation>
      ...
    </author>
  </authors>
  <description>
    <abstract>The deformation retract soap film ...</abstract>
    <detail> ... </detail>
    <msc2000>
      <primary>49Q20</primary>
      <secondary> ... </secondary>
    </msc2000>
    ...
  </description>
  <files>
    <file_master format="jvx">
      <filename>Retract_Master.jvx</filename>
    </file_master>
     ...
  </files>
</eg-model>
```

Fig. 4. Shortened version of the XML description of model 2001.01.042.

If you are viewing our model on the Web, you do not see the XML file listed in Figure 4. Instead you see an HTML file which has been obtained by applying an *XML transforming style sheet* (XSLT) to the XML document. What is the advantage of having an XML file plus an HTML file which was just mechanically produced from the former? Actually, there are several.

As already alluded to above, the XML document itself is free of style elements such as font style and color and such. If this is bound to change one can easily adapt one transformation style sheet and apply it to thousands of XML documents, without touching the documents themselves. But, this is only a minor advantage. More recent versions of HTML allow for *cascading style sheets* (CSS) which provide for exactly this. There are other advantages for the XSLT.

Many of our models are geometric objects which live in Euclidean 3-space. In order to offer the users of the Server better geometric insight, these models are interactively displayed with the applet version of JavaView. Java is now widely spread, but on some operating systems

```
<xsl:template match="eg-model">
<eg-zbl>
   <xsl:apply-templates select="eg-id"/>
   <xsl:apply-templates select="title"/>
   <xsl:apply-templates select="modelname"/>
   <xsl:apply-templates select="authors"/>
   <xsl:apply-templates select="description"/>
</eg-zbl>
</xsl:template>
<xsl:template match="description">
  <description>
    <xsl:apply-templates select="abstract"/>
    <xsl:apply-templates select="msc2000"/>
    <xsl:apply-templates select="keywords"/>
    <xsl:apply-templates select="zentralblatt"/>
  </description>
</xsl:template>
<xsl:template match="*|@*">
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:apply-templates select="node()"/>
  </xsl:copy>
</xsl:template>
```

Fig. 5. Transforming style sheet to produce Metadata information for Mathematisches Zentralblatt.

Java installation is still far from being perfect. Moreover, some users with a slow connection to the Internet might be willing to save loading the, albeit small, applet, and the preview model if they are provided with just a single image. We took care of that. So there are two HTML pages for each model: One with the applet, one without (the *preview* version). The user can easily switch back and forth between the two kinds of pages. This is where XSLT comes in. Just one XML file per model plus two transformation style sheets for all the models together makes two HTML pages per model. This goes way beyond what is possible to express with CSS.

Given the fact that there are actually even more views on one model which have to do with the reviewing process, the benefit of this approach is even much larger. In order to give an idea see Figure 5 for the transformation style sheet which produces a new XML file from the model description XML, which basically contains Metadata information. In this case the resulting XML file is sent to Mathematisches Zentralblatt for reviewing. A transformation style sheet works along the rooted tree of the XML document essentially in a depth first way. At each node it applies pattern matching in order to find out how to proceed. In this case it exactly copies part of the original document. The style sheet itself is valid XML. Note that it is allowed to abbreviate an empty node `<tag></tag>`

by `<tag/>`. The aforementioned style sheets for producing the HTML files are very similar in flavor, just much more elaborate.

We still have not yet arrived at the most striking advantage in favor of XML. Setting up a server like EG-Models requires an enormous joint effort. And this is true for the few people who actually run the server only to a lesser extent. The server is worth nothing without the contributions of the authors. They often invest a lot of time in order to make their models perfect. It would be shameful if their models become useless at some time in the (possibly near) future when HTML and maybe even XML will be gone. If HTML dies out, then there will be something else (probably XML itself) to replace it. Again changing the transformation style sheets to make them produce whatever is up to date then solves the problem. But, what if, or rather when, XML itself will die out? The solution is at hand: There will be just one final XSLT which will look very similar to the one in Figure 5 which does the job and transforms the XML document into the format of the future without loosing a bit of information. From the point of view of the EG-Models Server the — by far — most crucial point about XML is its built-in capability of getting rid of itself.

5.2 No Need to Know about XML

XML might a look a bit awkward at first sight. So it is important to stress that nobody is forced to actually *type* XML by hand. This is true for both the textual model description and the geometry (JVX) file. An author may avoid typing XML by choosing one of the following methods:

1. The description of a model including author information, literature etc. may be entered in a submission form available on the EG-Models Server. This interactive dialog allows typing the content of the submission description in plain text and saving the data as an XML formatted file. As a convenient choice, the dialog offers loading an existing description template of the author which may already contain basic information such as the author's address.
2. JavaView itself can be used to produce such descriptions. The command "Submit EG-Model" in the "File" menu offers a dialogue similar to the one one the server, basically with the same functionality. This way, it is possible to produce XML descriptions without being connected to the Internet.
3. The geometry file of a model may be a non-XML file format like BYU or OBJ, although we strongly recommend using the XML based file format JVX. Other file formats may be converted into JVX using JavaView.

As already mentioned, the possibility for the automatical validation of data files is crucial. On the server we provide a service to authors to check a potential submission for syntactical correctness. This includes an option to preview a submission in the style it would look if published on the server.

6 Access and Copyright of Published Models

An import aspect of archived data is an easy and direct access to the data, and a simple citation and reference mechnism. Many of the drawbacks of classic print journals disappear in the realm of electronic versions. For example, it is possible to have a direct web link to an individual publication, or to allow a full-text keyword search.

6.1 Unique Identification Number

Each model submission receives a unique identification number if the uploaded submission is complete and on first sight follows some basic requirements. Even rejected models which fail the later refereeing process will still keep their identification number for internal archiving purposes. Any published model on the server can uniquely be cited using this identification number.

The identification number has the form *xxxx.yy.zzz* where the four digits *xxxx* are the year of submission, the two digits *yy* the month of submission, and the three digits *zzz* are an internal counter of the received submissions starting with *001* each month and incrementing with each model. For example, the retract of Brakke has identification number *2001.01.042* which is the *42*-nd model received in January 2001.

The identification number is also a key part of a unique URL which enables a direct internet access to a published model. Simply, attach the identification number to the EG-Model domain as in the following example:

http://www.eg-models.de/2001.01.042/

The EG-Models data base stores further keywords which allow providing various other ways to view and access the model collection. For example, a directory like browsing is based on a hierarchical set of subject keywords, or search robots may find models based on different search criteria.

6.2 Copyright

The specific form of publication on the EG-Models Server requires a special copyright concept. While EG-Models seems to be in a similar position as the publisher of a journal, the authors usually want to use and exhibit their models after publication. The following copyright transfer statement tries to take into account both positions.

> Submission of a model implies that the model has not been published before, except perhaps in the form of an abstract, part of a published lecture, review, or thesis; that it is not under consideration for publication elsewhere; that its publication has

been approved by all authors and (if appropriate) by the institution where the work was carried out; that, if and when the model is accepted for publication, the authors agree to an automatic transfer of the non-exclusive right to exhibit and publish the master models and supplemental material within this model server on any media today and in the future to the publisher of the server; and that the master model will not be published elsewhere in any format without the consent of the copyright holders.
The authors retain all other copyrights, in particular the right to publish any image of the model elsewhere. Nevertheless, the public exposition of information requires a copyright transfer that provides the legal basis such that the publisher of this archive may exhibit and publish the digital models and the accompanying data on the server and on future publication media.
Whenever the publisher of the archive exhibits or publishes models of the archive then authors of the models are cited and identified as authors of the models.

7 Summary and Outlook

We have given an in-depth overview of the structure and implementation of a peer-reviewed, fully electronic journal for the publication of digital geometry models. We develop quality criteria and XML formats for digital geometry models and experimental data sets. The internal documents are all XML based which allows the automatic validation of geometry models and explanatory texts, the automatic generation of different web presentations, and a simplified editorial communication. Many of the concepts and tools developed for the EG-Models server are not specific to the content of our server but extend to the management of other types of scientific electronic journals.

8 Samples

This section presents some models published on the EG-Models server to give an overview of the variety of mathematical topics and to discuss various technical and computational aspects.

Deformation Retract Minimal Surface This model [1] shown in Figure 6 by Ken Brakke analyses the tricky question of exactly what minimization problem a soap film solves. Reifenberg considered soap films to be sets that spanned a boundary in a homological sense. Adams described this soap film as an example of a set we would like to call a soap film, but which does not span its boundary in any homological

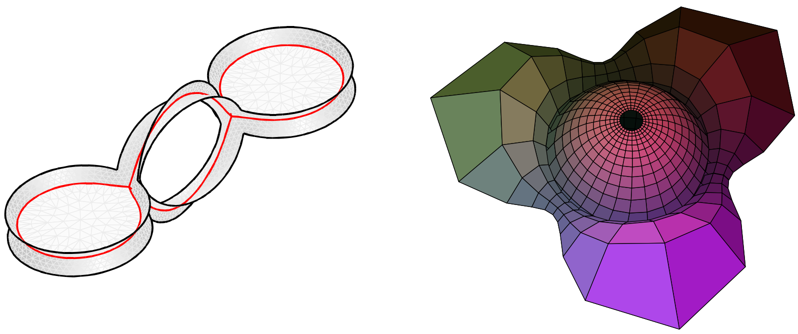


Fig. 6. Deformation retract minimal surface (left) and the Darboux transform of a discrete spherical isothermic net (right).

sense. This film, on an unknotted wire, has a deformation retract to the wire. A retract from a set Y (the film) to a subset X (the wire) is a continuous map from Y to X which is the identity map on X. A deformation retract of Y to X is a homotopy from the identity map on Y to a retract of Y to X, with the image of the homotopy remaining in Y. It is, in fact, a strong deformation retract since the homotopy can be chosen to always be the identity map on the wire. The existence of this deformation retract shows that the soap film cannot be said to span the wire in any homological sense (at least if one stays with a wire-embedded-in-Euclidean-space model of soap films).

The data set of the model was produced by the Surface Evolver [2], and exported in the JVX format. Additionally, this model publication is accompanied with an interactive Java applet showing the actual homotopy as a user-controllable animation. Such additional interactivity may be supplied by an author by adding files in the category FILE_OTHER. These data belongs to the reviewed model submission, but the EG-Models managers provide no guarantee that these files are maintained. In the worst case, files in this category may become obsolete or may even stop working, for example, if a certain version of Java is no longer supported by major browsers.

In contrast, the files in all other categories will be maintained by the managers of the EG-Models archive. For example if the specification of XML changes at some future time, then affected files of EG-Models publications may be updated into a different format.

Darboux Transform of a Discrete Spherical Isothermic Net Smooth surfaces of constant mean curvature 1 in hyperbolic space can be characterized by the fact that a suitable Darboux transform (by means of the conformal Gauss map) yields the hyperbolic Gauss map. This provides one (of at least two) possibilities to define discrete horospherical nets – as analogs of smooth cmc-1 surfaces in hyperbolic space – as special discrete isothermic nets: note that the hyperbolic Gauss map, being part of the definition, determines the hyperbolic geometry the surface is horospherical in as a subgeometry of Moebius geometry. The displayed model [9] shown in Figure 6 by Jeromin was obtained as a Darboux transform of a spherical discrete isothermic net with high symmetry. It therefore is a horospherical net, and can be considered as a discrete analog of a smooth surface of constant mean curvature 1 in hyperbolic space: in the picture, the sphere at infinity of hyperbolic space sits inside the surface (the surface having two ends) – the standard Poincare ball model of hyperbolic space is obtained by inverting the configuration at the infinity sphere.

Authors often use their own inhouse tools or rely on various commercial software for the numerical computation. For example, this model was calculated with a Mathematica [19] program. We provide a set of filters to help with the conversion of a data set in the file formats supported by EG-Models.

Often the file format conversion must be accompanied with non-trivial operations on the mesh. For example, to calculate mesh adjaceny information, to simplify large datasets, to remove multiple vertices or cracks within the mesh. Some of these operations are beautifications but others are necessary to provide a well-defined, conforming mesh.

Simple Algebraic Singularities This collection of algebraic surfaces [12] by Richard Morris presents the simplest types of singularities which can occur for functions from $\mathbb{R}^3$ to $\mathbb{R}$. The types consist of two infinite sequences and three special cases. The zero sets of the simplest of these singularities are presented here. All models have been produced by an applet based on the JavaView API [14] and a server based program which calculates the zero set of a given function. The server is adapted from the program in the LSMP package. The numerical algorithm has been constructed to try to get accurate representations of the singular points in the surfaces.

For the purpose of this collection the models have been cleaned up by hand to give topologically accurate representations of the singular points. These files have been hand edited to ensure that the boundaries are correct and that they are topologically correct around the singular point.

Densest Lattice Packing of a Truncated Dodecahedron For a given convex body the lattice packing problem is the task to find a lattice of minimal determinant such that two different lattice translates of the body have no interior points in common. The ratio of the volume of the body to the determinant of such an optimal lattice is called the density of a densest lattice packing and it can be interpreted as the maximal proportion of the space that can be occupied by non-overlapping lattice translates of the body. The example in Figure 7 shows a truncated

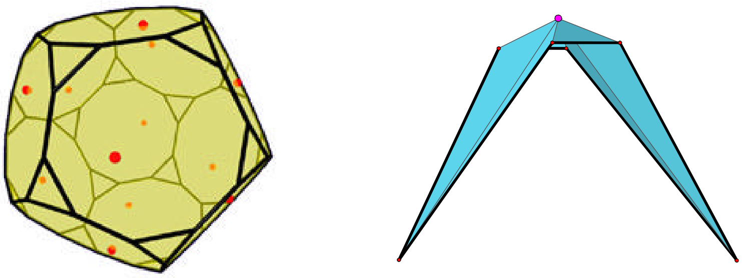


Fig. 7. Densest lattice packing of a truncated dodecahedron (left) and a counterexample to the maximum principle of discrete minimal surfaces (right).

dodecahedron [7] from a collection of models by Martin Henk where the density of a densest lattice packing was calculated with the algorithm of Betke and Henk. The density is equal to $(37+5\sqrt{5})/(24\sqrt{5})$, and the 12 points in the picture show the lattice points of a critical lattice lying in the boundary.

Counterexample to the Maximum Principle of Discrete Minimal Surfaces The authors define discrete minimal surfaces as piecewise linear continuous triangulated surfaces that are critical for the area functional with respect to all variations through surfaces of the same type that preserve the simplicial structure. Unlike smooth minimal surfaces, a discrete minimal surface might not lie in the convex hull of its boundary, see Figure 7. This example [15] by Konrad Polthier and Wayne Rossman is a simple case of such a surface, and thereby disproves existence of a discrete version of the convex hull principle for discrete minimal surfaces.

More general, the maximum principle for solutions of elliptic partial differential equations says, that if two solutions are tangent at a common point and one solution lies on one side of the other solution then both solutions are identical. This maximum principle does not hold for discrete minimal surfaces as this model demonstrates: the central vertex lies outside the convex hull of its boundary, therefore, any xy-planar triangulation, which is certainly discrete minimal, lies one side of the example. For example, consider the planar triangulation obtained from projecting the surface onto the xy-plane.

Sharir's Cube A three-dimensional convex polytope is a *cube* if its face lattice is isomorphic to the standard cube, which arises as the convex hull of all three-dimensional vectors with 0/1-coordinates. Many examples of strange cubes are known. Some of them turn out to be surprisingly difficult to handle for certain (Simplex type) algorithms for linear programming. Each quadrangular facet F^+ of a cube has an *opposite* facet F^-,

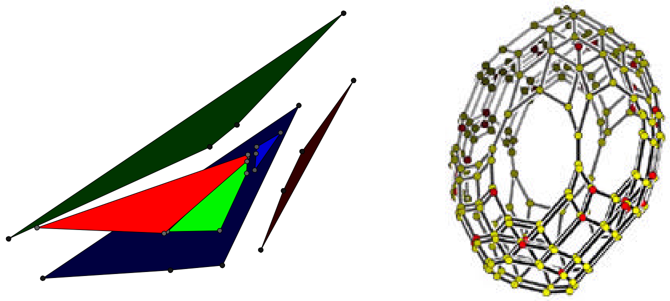


Fig. 8. Explosion of the boundary of Sharir's Cube (left) and a secondary polytope with interior points (right).

which is characterized by the condition that F^+ and F^- do not share any vertex. In the standard cube any two opposite facets are parallel, but it is easy to deform a cube such this is no longer true. Micha Sharir asked the question whether it is possible to have a cube such that each facet is *perpendicular* to its opposite. Ziegler gave an explicit construction [21] which establishes a positive answer. The model is shown in Figure 8.

For this kind of model it is essential to have exact data. Therefore, the master file is in polymake format, which allows arbitrary precision rational coordinates. Any representation of a polytope with floating point coordinates which fail to satisfy the numerical conditions by a small margin would be entirely useless.

A Secondary Polytope An ingenious construction of Gelfand, Kapranov, and Zelevinsky [5] associates to each triangulation of a set of n points in d-space one point in n-space. The i-th coordinate of this new point is the total volume of all the simplices incident to the i-th point in the triangulation. The convex hull of these new points is called the *secondary polytope* of the point configuration. It has dimension $n - d - 1$, and its vertices correspond to the so-called regular triangulations of the point set.

Since it is a difficult task to enumerate all the triangulations of a given point configuration, it is computationally hard to produce non-trivial examples of secondary polytopes. Pfeifle used TOPCOM [16] and polymake to provide some examples, see [13] and Figure 8, which can now be used as a starting point for further investigations.

References

1. K. Brakke. Deformation retract soap film. *Electronic Geometry Models*, 2001. http://www.eg-models.de/2001.01.042/.
2. K. A. Brakke. The surface evolver. *Exp. Math.*, 1(2):141–165, 1992.
3. G. Fischer. *Mathematische Modelle/Mathematical Models*. Vieweg, Braunschweig, 1986.

4. E. Gawrilow and M. Joswig. polymake: an approach to modular software design in computational geometry. In *Proceedings of the ACM Symposium on Computational Geometry*, 2001. 10 pages, to appear.
5. I. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, resultants, and multidimensional determinants.* Mathematics: Theory & Applications. Birkhäuser Verlag, 1994.
6. K. Grosse-Brauckmann, B. Oberknapp, and K. Polthier. Penta - a constant mean curvature surface of genus. *Electronic Geometry Models*, 2000. http://www.eg-models.de/2000.09.039/.
7. M. Henk. Densest lattice packing of a truncated dodecahedron. *Electronic Geometry Models*, 2001. http://www.eg-models.de/2001.02.064/.
8. D. Hoffman and W. Meeks. Complete embedded minimal surfaces of finite total curvature. *Bull. AMS*, pages 134–136, 1985.
9. U. Jeromin. Darboux transform of a discrete spherical isothermic net. *Electronic Geometry Models*, 2000. http://www.eg-models.de/2000.09.038/.
10. M. Joswig and K. Polthier. Digital models and computer assisted proofs. *EMS Newsletter*, December, 2000.
11. M. Joswig and K. Polthier. Digitale geometrische Modelle. *DMV Mitteilungen*, 4:20 – 22, 2000.
12. R. Morris. Singularities of algebraic functions in R^3. *Electronic Geometry Models*, 2001. http://www.eg-models.de/2001.06.002/,..,2001.06.020/.
13. J. Pfeifle. Secondary polytope of a cyclic 8-polytope with 12 vertices. *Electronic Geometry Models*, 2000. http://www.eg-models.de/2000.09.032/.
14. K. Polthier, S. Khadem-Al-Charieh, E. Preuß, and U. Reitebuch. JavaView Homepage, 2001. `http://www.javaview.de/`.
15. K. Polthier and W. Rossman. Counterexample to the maximum principle for discrete minimal surfaces. *Electronic Geometry Models*, 2001. http://www.eg-models.de/2000.11.040/.
16. J. Rambau. Topcom 0.9.0. http://www.zib.de/rambau/TOPCOM/.
17. M. Schilling. *Catalog Mathematischer Modelle Für Den Höheren Mathematischen Unterricht.* Leipzig, 1911.
18. H. A. Schwarz. *Gesammelte Mathematische Abhandlungen.* Springer, Berlin, 1890.
19. Wolfram Research. Homepage. `http://www.wolfram.com`.
20. G. M. Ziegler. *Lectures on Polytopes.* Springer, 1998. 2nd ed.
21. G. M. Ziegler. Sharir's cube. *Electronic Geometry Models*, 2000. http://www.eg-models.de/2000.09.028/.

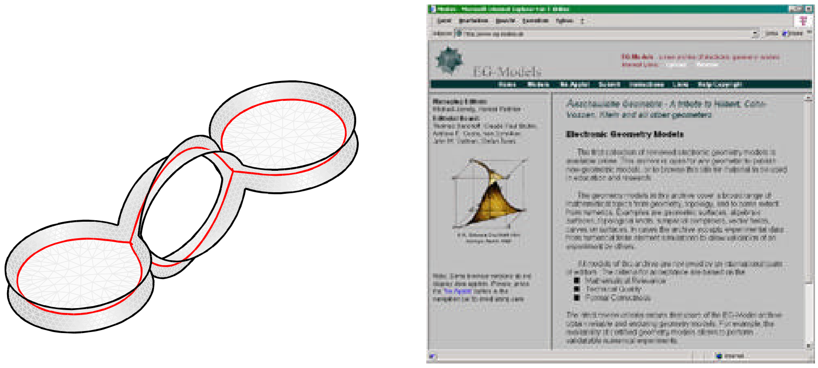


Fig. 1. Deformation retract minimal surface (left) and Homepage of the new EG-Models server (right).

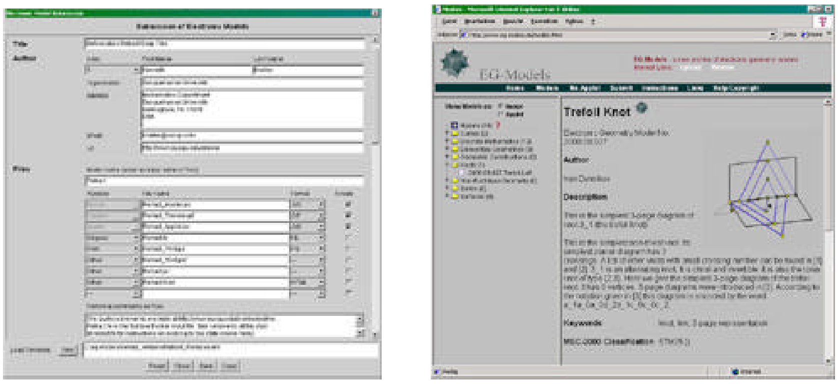


Fig. 2. Online submission form and a published model (right).

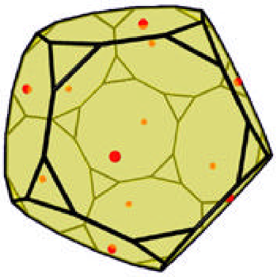
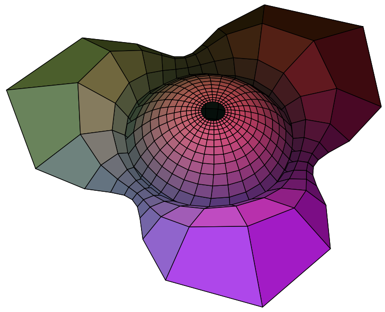
 

Fig. 3. Densest lattice packing of a truncated dodecahedron (left) and the Darboux transform of a discrete spherical isothermic net (right).

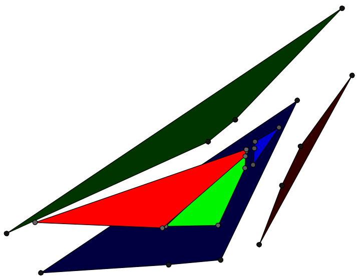
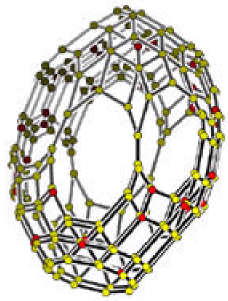
 

Fig. 4. Sharir's cube (left) and a secondary polytope (right).