Prospects of LGS: Application and Development

Sergey Preis[1], Alexey Kazakov[1]

[1] LEDAS Ltd, 6 Prospekt Lavrentieva,
Novosibirsk Russia 630090
spr@ledas.com, lex@ledas.com
http://www.ledas.com

Abstract

The paper presents LGS, a geometric solver by LEDAS for systems requiring solving of geometric problems with constraints. LGS is presently a stable marketable product, and so the paper focuses on the future of LGS, such as possible development paths and applications to real tasks. Apart from the traditional use in large parametric CAD systems, possibilities for integrating the solver in small CADs and three-dimensional modeling systems are examined. Special attention is paid to applications of engineering constraints for geometrical models.

Keywords: *CAD, intelligent functions, geometrical constraints, engineering constraints, geometrical solver.*

1. LGS GEOMETRIC SOLVER

Large CAD systems have long been using tools for construction and modification of drawings that do not require the user to literally draw all the elements and parts. For an advanced CAD system, it is enough to specify a sketch, indicate the geometric and other relationships between its parts that must be observed in this project, and enter some fixed or desired values of individual elements. Based on this information and on built-in geometric knowledge, the Geometric Solver, a component of the CAD system, will generate all of the data needed for automatic creation of the drawing planned by the user. Within this approach modifications of a constructed drawing is performed via modifications of the relationships between its elements, geometrical solver updates a geometry of a drawing automatically. Thus, the cumbersome, complicated, and error-prone process of construction, development, and modification of a drawing is transformed into comfortable interaction with a high-performance, high-level expert system.

The Geometric Solver relieves the user of a significant amount of work and responsibility for the drawings, and dramatically increases the efficiency of the process due to the geometric knowledge accumulated in the solver and the ability to process the knowledge in accordance with the data for the specific project. It is the class of these solvers that LGS, a product developed by LEDAS belongs to; its first version was released in January 2004.

The current version of the two-dimensional LGS solver (LGS2D 1.2) is a reliable product able to compete equally with the leaders of the market of geometric solvers. It has almost everything that a user wishes to see in systems of this kind: LGS supports all the basic objects and constraints, arbitrary user-defined curves, symmetry constraints, equality of distances and radiuses and many other things. The last version of the LGS2D supports mechanisms that allow incorporating engineering knowledge into the geometrical model: black-box constraints, user-defined variables and equations. Not only it allows one to find a predictable solution of a constraint problem with performance sufficient for interactive operations even if the model is under- or overconstrained; it also supports modification of the drawing via translation and rotation of objects and can output the set of constraints that make the problem overconstrained.

The information on the LGS solver can be found at http://lgs.ledas.com. One can also download LEGE'n'D, a demo application that illustrates the main capabilities of LGS.

In the first part of the paper, we will describe the history, current state of the LGS and the functions added to its latest version. Then we will try to present possible paths for further development of LGS. We will cover a few most promising areas for extension of the solver and define the tasks, where these extensions may be applied. We will also discuss in more detail the applications where LGS may be employed, trying to go beyond the traditional applications of variational solvers and see what other system may benefit from a geometric solver and what problems it can solve.

2. LGS: PAST AND PRESENT

It is two years now since the first preliminary version of LGS was made available to a large group of users as part of an early access program (EAP) organized by LEDAS. For two years the company has been observing the considerable interest attracted to the solver. More than 100 users from the US, Japan, China, Israel, France, Ukraine, and Russia participated in EAP and paid attention to demo application after the public release of version 1.0. In accordance with the company's schedule and the suggestions of potential customers the system was developing, the set of functions was expanding, performance was improving, behavior was becoming more natural, and the test base was growing. The development of LGS from April 2003 to March 2005 may be characterized by table, which reflects the development of the LGS solver in terms of its functions.

The table shows that LEDAS has done a great deal of work over the last two years. Apart from developing and testing the solver proper, the team has created a demo application LEGE'n'D based on LGS and OpenCASCADE (downloadable from http://lgs.ledas.com/download), a trial integration of LGS with bCAD from ProPro was performed, and several papers were published in the leading Russian publications covering CAD research. This confirms that LGS is a ripe product, which is also confirmed by sales and interest it attracts from the public.

However, LEDAS is not pausing to rest yet. The scheduled development of LEDAS intends to broaden significantly the range of applications for the solver and turn it into one of the most powerful existing solutions of this kind. The nearest plans include implementation of support for inequalities and improvement of the diagnostics methods for geometrical model, including support for degree of freedom analysis and well- and underdefined diagnostics.

At the same time, LEDAS sees its mission in bringing these state-of-the-art parametrical capabilities to most users. The new LGS capabilities will be available in a broad range of systems, from entry-level CAD systems and simple 3D-modeling software to specialized packages.

Table 1. Dynamics of LGS functions

Features	LGS 0.3 Mar 2003	LGS 0.5 Jul 2003	LGS 0.9 Dec 2003	LGS 1.0 Jan 2004	LGS 1.1 May 2004	LGS 1.2 Mar 2005
Objects						
Points, Lines	+	+	+	+	+	+
Circles	+/-	+	+	+	+	+
Ellipses	-	+	+	+	+	+
Rigid sets	-	+	+	+	+	+
Parametrical curves	-	-	+	+	+	+
Variables	-	-	-	-	-	+
Constraints and dimensions						
Incidence, Distance, Angle, Parallelism	+	+	+	+	+	+
Radius	-	+	+	+	+	+
Tangency	-	+/-	+	+	+	+
Symmetry, equal distance	-	+/-	+	+	+	+
Dependency	-	-	+	+	+	+
Equations	-	-	-	-	-	+
Black-box constraints	-	-	-	-	-	+
Functions and features						
Calculate	+	+	+	+	+	+
Move, rotate	-	-/+	+/-	+/-	+	+
Indication of overconstrained areas	-	-	+/-	+/-	+	+

Knowledge about engineering properties of the product is often related to some of the product's parameters, which, in turn, depend on the geometrical properties of the product, such as area or center of mass. The newest feature of the LGS2D 1.2 version is the support for variables, equations and “black-box” constraints that allows one to link engineering properties with the geometrical parameters of the product.

The simplest way to associate engineering properties with the model is to impose additional algebraic equations for parameters of the geometrical constraints. These equations may express different properties (such as material consumption or mass of the product) that can't be expressed via ordinary geometrical constraints. For example, constraint of a length of a contour that consists of segments and circular arcs can be modeled via algebraic equation linking parameters of distance constraints, parameters of angle constraints and radii of the arcs. In the example on Fig. 1 such equation is used to model the length of the belt transmission between two pulleys.

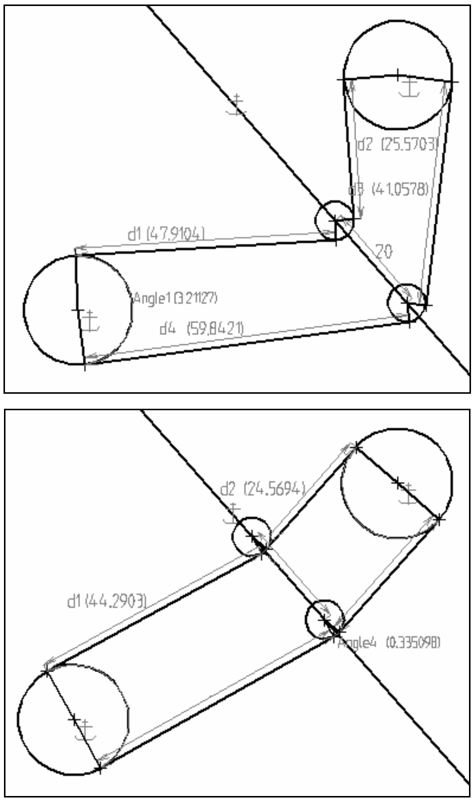

Fig. 1 Constraint of length of the contour

In this example, two small pulleys lying on the fixed line are used to provide tightness for the belt. If one modifies a length of the belt the new position of the pulleys on the line will be calculated. And vice versa, if the length of the belt is used as reference constraint it will be updated when pulleys are moved along the line. Another examples are the ability to link the circumference of a circle with the length of the segment it is rolling on, or to define the radii of gears in accordance with the gear ratio.

More complex engineering constraints can be introduced via more general mechanism of "black-box" constraints provided by LGS2D. This mechanism allows specifying a constraint for geometrical objects and parameters with the semantics fully defined by an application. Application just provides a function, which calculates a discrepancy of the constraint by given positions of its geometrical arguments. The function is called when LGS2D solves the constraint. The application's constraints are treated equally with the built-in ones, and so cyclic dependencies involving constraints and geometric objects are allowed. The simplest examples of usage of the "black-box" mechanism are implementation of the area and length of the contour constraints.

3. LGS TRENDS: KNOWLEDGE, OPTIMIZATION AND MORE

Current LGS functionality is quite sufficient for the typical problems submitted to the geometric solver by modern CAD systems. Indeed, LGS can easily handle construction of an exact model from a rough sketch with constraints, as well as analysis and subsequent modification of the model thus constructed. It is these possibilities that are usually used in feature-based design in the models for creation of planar drawings, rigid-body simulation and assembly models. However, LEDAS is targeting intelligent tools for PLM, including CAD, which means inclusion of diverse knowledge in CAD models. Creation of a really intelligent solver is the strategic development vector. LEDAS analysts define the following possibilities for LGS development:

- three-dimensional solver for problems about assembling and engineering parts and mechanisms, as well as for many other applications. The work has begun, and the first version of LGS3D is available since November 2005 within early access program (EAP). Subsequently this functionality will be expanded to intelligent applications, and all of the topics below apply to both versions of the solver, two-dimensional and three-dimensional;
- "soft" constraints as a tool for specification of knowledge on the nature of constraints and for modeling of spring connections;
- optimization by given criteria as a way to solve a wide range of application problems related to improving a product's features;
- analysis of degrees of freedom in a product as a way to study mechanical properties of a product and correctness of its operation;
- automatic addition of necessary constraints to an underconstrained model, in order to simplify design and data exchange with systems that do not support constraints, or using hard copy;
- computation of interval estimates for possible values of parameters as a means to analyze configurations, to support tools for collaborative design and exchange of knowledge.

These topics fall into four major areas that compose the perspective development plan for LGS. Let us examine them in more detail.

3.1 Soft constraints and optimization

Really important and demanded possibility for enriching a model with extra knowledge about the product being designed is precise specification the nature of links, or relationships between its parts and components. Moreover, this information can be used to optimize the shape of the product according to a criterion. The support for specifying constraint properties is the second promising vector for LGS development.

The only property that should be maintained for a constraint is a certain description of its stiffness. A typical example is the distance constraint for a spring connection: the distance has a desired value, but it can be slightly larger or smaller. During calculation, one needs to determine the positions of the components that minimize the total deviation from the desired values - which is the natural optimization criterion for problems with springs. However, the target functions may be diverse: deviation from the desired distance for a spring, length of a loop, etc. In addition, one must be able to calculate the deviation from the desired value for each spring constraint. The problem becomes slightly more complex if we introduce the elasticity modulus, i.e., the parameter that defines which spring is "easier" to extend or compress.

Soft constraints can be used in a very broad context: they may determine manufacturability of a product, material consumption, etc. In the general case, elasticity modulus is similar to weights or priorities of constraints, while the target functions may be exactly the same: minimal deviation from desired values, minimal area of the product, etc.

It is assumed that for measure constraints (distances, angles), as well as for the constraints of incidence and tangency (which will be defined by distance), the user will be able to specify the weight of a constraint, i.e., how desirable it is, as well as the target parameter for optimization. LGS will solve the optimization constraint problem and calculate the deviations of the parameters of "soft" constraints from the desired values (calculate the residuals of these constraints).

3.2 Degrees of freedom analysis

The notion of degree of freedom plays significant role in the construction of geometrical drawings and models, because it's very important to know which parts of the model can be moved or rotated and which cannot within given configuration of geometrical constraints.

In three-dimensional case for an assembly model containing several rigid parts joined by assembly constraints the knowledge of present degrees of freedom allows to analyze possible motions of the system and determine whether the product has desired mechanical properties.

In two-dimensional case the parts of the sketch that has no internal degrees of freedom (i.e. well-defined parts) can be modified via modifications of parameters of constraints and it's guaranteed that the changes will be fully controlled by constraints parameters and hence by user himself. The diagnostics of such parts allows creating correct sketches that can be modified in a predictable way.

Thus, the functions expected from a geometrical solver are

- computation of rotational and translational directions of instantaneous motion for parts of a geometrical model
- identification of well-defined parts of a model, i.e. parts without internal degrees of freedom

The first of these functions relates rather to assembly process in the three-dimensional case, the second relates to the construction of well-defined parts of two-dimensional sketches. But, this separation isn't so strict because the exact computation of degrees of freedom can be required by two-dimensional applications, e.g. for computation of possible motions of a planar kinematical model. And vice versa, the identification of the well-defined parts may be useful during construction of a three-dimensional model.

3.3 Automatic constraining

The third vector for future development of LGS targets making the solver more intelligent, i.e., providing it with additional knowledge about the nature of constraints. Based on this knowledge, we propose to implement the function that constructs the necessary (and sufficient) constraints for an underconstrained model or its part.

To be more precise, the task is, given a set of pre-positioned geometric objects linked by a small number of constraints, or none at all, to find a set of constraints that would make the model rigid, i.e., lacking internal degrees of freedom. An important aspect is that the constraints imposed must be natural in some way. This requirement implies that a rather large amount of knowledge must be incorporated in LGS in order to solve the automatic constraint specification problem.

However, we believe that the complexity of implementation will be rewarded, because automatic constraint specification is a badly needed function in various CAD-related systems. It can accelerate the construction of user-defined drawings by reducing significantly the number of constraints that the user must specify manually. In data exchange systems it can help to transmit data from nonparametric CAD systems to parametric ones. The same function may be used by the systems that digitize hard-copy drawings in order to convert the existing drawings to a vector-parametric form. In addition, this function may be used far beyond the limits of CAD software, for instance, to discover regularities in arbitrary geometric information, for instance, in bioinformatics.

3.4 Interval solver

An interval solver could be very useful as a part of a collaborative design system in CAD/PLM. When working on a product together, it is important to supply the partners with as much knowledge about the product as possible, and so transmitting intervals of allowed values for parameters produces better results than exchanging exact values only. An interval solver could calculate the intervals of allowed values from the geometric constraints and thus could help in collaborative design. However, the lack of an environment in which it could be used efficiently reduces the priority of this task compared to the ones above. This priority may be increased if our partners request interval functionality.

4. LGS APPLICATIONS: BEYOND FEATURES

Geometric solvers are most frequently applied in feature-based CAD systems. This is understandable: clear separation of the design process into phases, manipulating planar drawings, power of the systems ensure that the solver can be used effectively in many of such systems. Below we discuss several extensions of this traditional field.

4.1 Extending classic applications

Planar drawing construction (sketching) systems are typical applications for two-dimensional solvers. Since the information about the base planar drawings is preserved by these systems over the entire design period, the use of a solver allows reducing the amount of data entered and ensures simplicity of modification of the product by means of changing the planar contours even in the latest phases of design.

Three-dimensional solvers are most frequently used for rigid-body design and construction of assembly models. They allow both defining the geometry of parts using parameters and specifying dependencies between the parts according to their relative location (assembly drawing).

However, the applications of the solver may be much broader. Thus, support for parametric user-defined curves may help in the processing of models constructed on curve meshes, support for user-defined parameters may allow simulation and analysis of physical and economic properties of the product, while optimization will allow finding the best solutions from the viewpoint of manufacturability.

4.2 Integration in small CAD systems

An overwhelming majority of feature-based CAD systems belong to the high-end or upper-middle level, but the low-end systems and three-dimensional simulation systems use exactly the same geometric information, and so potentially can benefit from the use of the solver. Nevertheless, due to high cost of solvers and some ideological problems, the number of parametric desktop CAD systems and three-dimensional design systems is very small. A pleasant exception is T-FLEX CAD from Top Systems, a fully parametric CAD system containing a proprietary solver.

The method used in T-FLEX to construct planar parametric drawings is rather different from variational design implemented in the two-dimensional drawings of large CAD systems. In the latter case, the drawing consists of graphic primitives (segments, arcs, etc.), which form the shape of the product; they are explicitly connected by geometric constraints creating a planar parametric model. A different approach is employed in T-FLEX: "First, the drawing is constructed in thin lines (construction lines), and then emphasized by the main lines (image lines). When doing the construction, you specify the relations between various elements (parallelism, perpendicularity, tangency, etc.) and fix the parameters, such as radius, angle, etc. By performing the procedure for building a drawing, you build a parametric drawing (http://www.topsystems.ru). Specialists of ProPro say that this procedure for construction of parametric drawings is simple, and natural for users of the small CADs, and is a very probably scheme for integration of LGS and bCAD, a small CAD system from ProPro.

Another possibility for using LGS in small CAD systems is to employ it for tracking links. These links have traditionally fixed the relative location of the elements of the drawing (for instance, one segment must intersect another exactly at the mid-point); instead of a strict specification of relative location, the system may define the corresponding constraints, which will make the drawing parametrical, and so more flexible.

When integrating the solver into small CAD systems, one must bear in mind the simplicity of these systems for the user and try not to add new functions, make the use of the solver transparent or at least natural for the user, i.e., try to make the solver similar to the tools the user is used to.

4.3 Three-dimensional modeling: applications for a 3D solver

The use of a full-scale three-dimensional solver certainly offers much greater possibilities. Unlike feature-based design systems, in the initial-level and three-dimensional modeling systems all the planar drawings are done in the three main views. When using a three-dimensional solver, one can specify not only constraints within a single view, but also constraints connecting elements lying in different planes. In addition, the constraints may be taken into account during subsequent transformations of the three-dimensional model, preserving the desired properties of the object of modeling.

The same constraints can be subsequently used to model the behavior of the object in motion (kinematics), as well as to calculate the phases of motion from the displacements of the key point (inverse kinematics). Unlike the classical solvers for inverse kinematics (IK-solvers), LGS can model the motion even if the skeletal model contains loops, and the use of soft constraints will allow it to emulate stretching of joints during complicated motions, spring connections, etc. The use of user-defined parameters may make motion even more realistic due to introduction of a physical component into the model.

For full-scale modeling of scenes in three-dimensional systems, however, another LEDAS solver is more appropriate: the physical simulation engine Phoenix that al-lows to calculate the motion of a system of rigid bodies with gravity, assembly connections (linkages), springs, and the condition of non-penetrating interaction with a provision for friction and heat emission. It is possible that Phoenix can calculate the motion of the background objects in a scene and take into account the movement of the key points of a character, while the animation of the character will be calculated by the geometric solver from the skeletal model.

5. CONCLUSION

It is clear that possible applications for LGS are not limited to the above areas. Wherever there are constraints on some geometric objects, a geometric solver can be used: in physical chemistry and molecular biology, computer graphics systems and desktop image processing, design of electronic circuits and main pipelines, and many other tasks. The development of LGS is also not limited to the vectors described in the paper: the solver can be customized for various problems and application areas.

LEDAS is ready to help everyone who is wishes to analyze applicability of LGS to a specific domain, or to adapt and integrate LGS for solution of specific problems. Our team will apply all of the knowledge and experience of the years of work in creating constraint solvers and building commercial systems on this foundation.

About the authors

Sergey Preis graduated from Novosibirsk State University in 1997 with specialization in computer science. He has been working at LEDAS company since its foundation in 1999. He is currently a system analyst at the LEDAS company in charge of products positioning and marketing plans. Also he takes part in several LEDAS projects as expert and advisor. His contact email is spr@ledas.com

Alexey Kazakov graduated from Novosibirsk State University in 2004 with specialization in applied mechanics. He has been working at LEDAS since 2001. Currently he is senior engineer in charge of developing of a new functionality in a geometrical solver project. He also participates in the project of physical simulation as an expert. His contact email is lex@ledas.com

LEDAS Ltd. is an independent software company founded in 1999 and located in Novosibirsk, Russia. LEDAS developed a state-of-the-art proprietary technology based on constraint programming, and applies it for PLM (Product Lifecycle Management) tasks, including CAD/CAM domain.