Automated Generation of Equations for Linkage Loci in a Game Physics System

Miguel A. Abánades

CES Felipe II
Universidad Complutense de Madrid
28300 Aranjuez
Spain

Phone: +34 918099200 ext. 253 • Fax: 918 099 205

mabanades@cesfelipesegundo.com

Francisco Botana

Departamento de Matemática Aplicada I
Universidad de Vigo, Campus A
Xunqueira
36005 Pontevedra
Spain

Phone: +34 986801931 • Fax: +34 986801907

fbotana@uvigo.es

Jesús Escribano

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madri
28040 Madrid
Spain

Phone: +34 913944679 • Fax: +34 913944654

jesus_escribano@mat.ucm.es

Abstract

A web-based tool designed to compute the equations of loci associated to linkages constructed using the game physics application Phun is presented. Complementing the graphing abilities of Phun, the equations are remotely computed using symbolic algebraic techniques from the field of Automated Deduction.

Besides adding exact mathematical knowledge to Phun, the algebraic completeness of the equations is shown to provide complete information about the different possible loci generated by a given mechanism in different configurations.

Special attention is given to four bar linkages motivated by the celebrated result by Kempe stating that any algebraic curve can be realized as the locus associated to a linkage composed basically of a combination of four bar linkages.

According to Abott, the number of links needed to generate an algebraic curve of degree n as the locus set associated to a linkage is $O(n^2)$. We show that the reciprocal relation between the degree of a linkage locus and the number of links does not exist.

The tool, named ALEPH (Automatic Locus Equation PHinder), has also educational applications as a bridge from game physics to (real) Mathematics.

Thought of as a remote add-on for Phun, its design consists of a simple web interface where the interaction with the user has been reduced to just one click. The tool is freely available at http://nash.sip.ucm.es/ALEPH where detailed instructions and examples of use can be found.

Keywords

Game Physics, Internet Accessible Mathematical Computation, Linkage , Dynamic Geometry, Computer Algebra, Locus

MSC codes: 97U60, 97U50, 51P05, 70B15

1 Introduction

We present a web-based tool designed to compute the algebraic equations of loci associated to linkages constructed using the game physics application Phun[1]. In this section we give a brief introduction to game physics and the relation between linkages and algebraic curves.

1.1 Game Physics Software

The relatively new term game physics refers to computer applications that offer an environment in which Newtonian laws are simulated, allowing the user to dynamically play with a virtual micro-world of simple objects.

With a core physics engine and complex collision detection algorithms as basic elements, game physics is a close approximation to real physics based on computations using discrete values. The most popular game physics application is Phun.

These dynamic applications can be viewed as the physics counterpart of the well-established Dynamic Geometry (DG) paradigm. DG systems are computer applications that allow the exact on-screen drawing of geometric constructions and their manipulation by mouse dragging certain elements. In fact the DG systems Cinderella (Richter-Gebert and Kortenkamp 1999), Geometry Expert (Gao et al. 1998) and The Geometer's Sketchpad (Jackiw 2002) include options for basic physical simulations similar to those of game physics.

[1] Freely available at http://www.phunland.com, multiplatform but closed code.

In an educational context, game physics software and DG systems are natural complements to each other. While game physics provides realism, Dynamic Geometry offers formal mathematical modelization. The application presented in this note is an example of this interplay: exact algebraic equations are computed for the trajectory of a point in a mechanism simulated with Phun.

But if there is something that makes game physics different from other educational computer applications is its level of acceptance among teenagers. A clear indication of this is the fact that despite being an emerging tool whose use in class is not nearly as widespread as that of dynamic geometry, the number of game physics videos in teenage popular YouTube[2] is significantly higher than that of those with dynamic geometry constructions. More concretely, while a search for the word "Phun" in YouTube gave 5240 results, the search for the word "GeoGebra" (one of the most popular DG systems[3]) gave only 792 results[4]. This popularity is without a doubt a great educational potential that teachers should exploit.

1.2 Linkages and Algebraic Curves

A mechanical linkage is a series of rigid links connected with joints to form a chain. Each link has two or more joints, and the joints have various degrees of freedom to allow motion between the links. Linkages are usually designed to take an input and produce a different output, altering the motion, velocity, acceleration, and applying mechanical advantage.

Linkages became an important research topic for mathematicians and engineers in the nineteen century after Watt's invention of the steam engine, which used a linkage to transform a cyclic movement into a straight movement (see Figure 1).

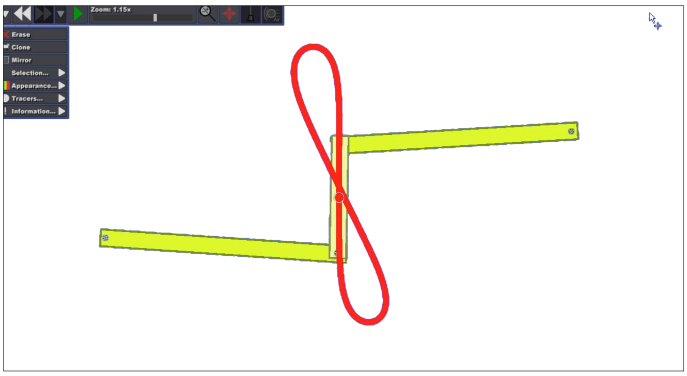

Fig. 1 Curve drawn by Watt's linkage in Phun

In fact Watt's linkage does not generate a straight path but an algebraic curve of degree six as shown in Section 3.

[2] http://www.youtube.com

[3] http://www.geogebra.org

[4] Search on April 20 2010

In 1876, Alfred Bray Kempe stated his Universality Theorem asserting that for any given plane algebraic curve a linkage can be designed to generate the algebraic curve as the locus set traced by a point in the linkage (Kempe 1876). Considering Weierstrass approximation theorem we have then that “there is a linkage that can reproduce your signature”, as Thurston put it (King 1999). In fact, Kempe's proof, like the one he gave for the famous four color theorem, was flawed but contained all key ideas.

Kempe's theorem settled an important question in engineering from the theoretical point of view, although the complexity of the mechanisms involved in his constructive description makes it impossible to use in practice. However, his study of linkages turned out to be useful in modern applications, such as CAD (computer assisted design) and robotics.

As practical implementations of Kempe linkage design, we have to mention the work by Gao et al. (2001) with the DG system Geometry Expert and that of Koebel (2008) with the DG system Cinderella.

These prototypes find a Kempe linkage for a given algebraic curve. What we present here is a web application that works the other way around. It provides the exact equation for the curve traced by a point in a linkage.

The problem of finding the equation for the locus set traced by a point in a geometric construction has long been a delicate issue in the field of Dynamic Geometry. In fact, no DG system is currently able to determine the equation of a general locus. The DG system Cabri implemented a first locus recognition algorithm but unfortunately no details are available. However, experiments show that the algorithm is relatively unstable. Figure 2 shows how Cabri wrongly assigns an equation of degree 3 to the locus set formed by a segment and half a circle.

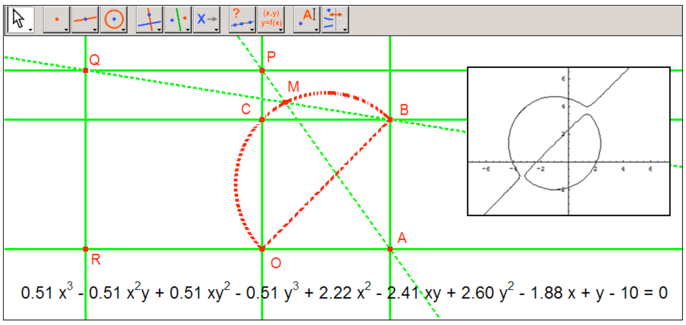

Fig. 2 Equation assigned by Cabri and the actual graph for the equation (small box)

This shows that approaching the recognition of computationally constructed loci by numerical means does not provide satisfying results. The authors have developed web applications5 that remotely obtain equations of loci constructed with The Geometer's Sketchpad, Cinderella, Cabri and GeoGebra based on more reliable symbolic algebraic techniques (Groebner bases) (Escribano et al. 2010). ALEPH follows the same ideas for linkage loci constructed in Phun.

---

[5] LADucation (http://nash.sip.ucm.es/LAD/LADucation.html) and LADucation for GeoGebra (http://nash.sip.ucm.es/LAD/LADucation4ggb/).

2 ALEPH: Automated Locus Equation Phinder

ALEPH is a web application that finds the equation of a linkage locus (i.e. a linkage with a tracer point or pen) constructed in Phun. It is a remote add-on for Phun.

The two basic guidelines in its developing process have been the simplicity of the user interface and the rigor of the mathematical methods behind it. To ensure user friendliness, ALEPH is just a web page with an upload button (see Figure 3). The soundness of the methods is ensured by its symbolic algebraic nature as described below.

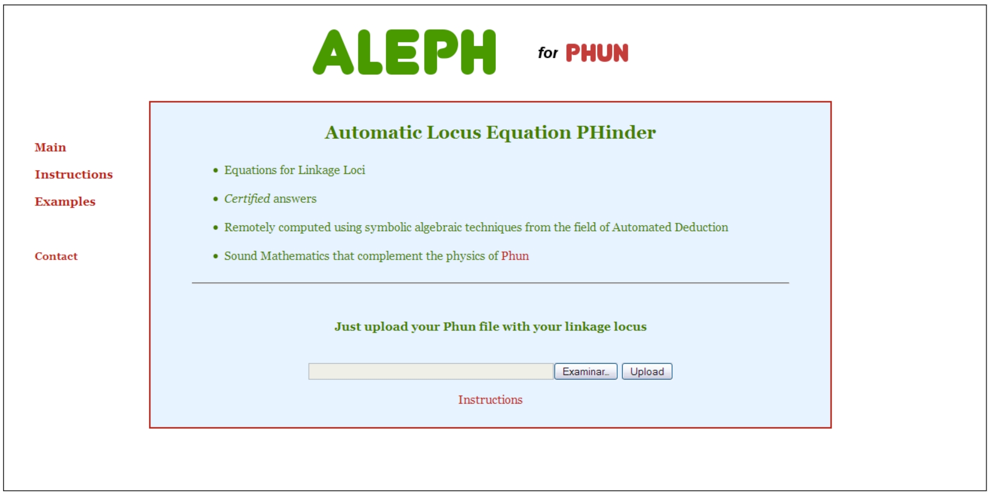

Fig. 3 ALEPH user interface as a simple web page

From the Phun own textual representation of a linkage locus construction (scene in Phun), a geometric description is created only in terms of points (hinges) and relations among these points. In fact, the geometric properties of a linkage can be described using only alignment conditions among points together with assignment of coordinates and distances.

As an example, if we consider the Watt's linkage in Figure 1, its locus can be geometrically described as

Points

175(-2.42,-0.35)

177(x[1],x[2])

178(3.33,0.62)

179(x[3],x[4])

180(u[1],u[2])

Conditions

distance(175,179)=2.53

distance(177,178)=2.53

Aligned(177,180,179)

distance(177,179)=1.17

distance(177,180)=0.57

distance(180,179)=0.6

Locus Point

180

Observe that the locus point (pen) is not on the exact middle of the link with hinges 177 and 179. This is due to the approximate nature of most Phun constructions. The numbers used as names for the points in the construction are those assigned by Phun in its internal description.

Note that, although the original Phun scene is described in terms of numerical data, symbolic coordinates are assigned to the points in the algebraic description of the locus. These symbolic coordinates (i.e. variables), together with the equations arisen from the conditions among the points, provide the algebraic setting of the problem in terms of polynomial ideals in the assigned variables. The algorithm used to obtain the locus equation is the one proposed by Botana and Valcarce (2003). Very roughly, the Buchberger algorithm is used to compute a Groebner basis for this ideal and obtain an elimination ideal. This elimination ideal is usually generated by one polynomial equation which is the algebraic equation of the locus.

To emphasize the differences between game physics and dynamic geometry, let us observe that the modelling of linkages in a DG system is generally not very satisfactory. Many DG systems, when dealing with intersections circle-circle they forget the intersection point they are working with when changing instances continuously. Special mentioned deserves Cinderella whose locus tracing algorithm is usually able to draw the complete loci even of complicated traces. Figure 4 shows Watt's locus as given by Cinderella and Cabri.

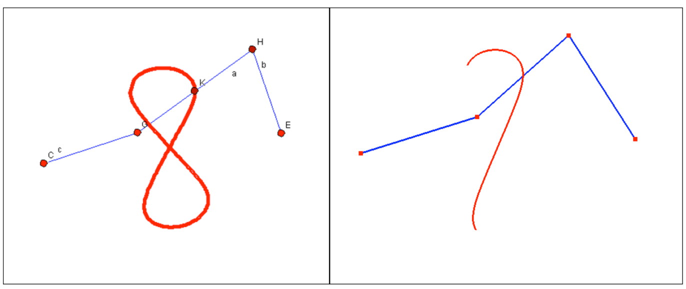

Fig. 4 Watt's locus by Cinderella (left) and Cabri (right)

Still, for this particular task (i.e. representation of linkage loci), we think that the most appropriate software is game physics.

With respect to the computational complexity of the algorithm, we have not analyzed it qualitatively but experimental results show that examples like the loci described by the feet of the walking linkages by Klann[6] and Jansen[7] are computationally out of reach. These limitations of the method are the algebraic counterpart of the limitations reported in other approaches to automatically recognize computationally (i.e. numerically) constructed loci (e.g. round off errors in Lebmeir and Richter-Gebert 2007).

2.1 Equation and Different Configurations of Mechanism

The algebraic nature of ALEPH makes its answers algebraically complete: all solutions of equations are considered. This has a geometrical interpretation: all points of intersection (line-circle, circle-circle,...) are considered. As shown above, a linkage mechanism has an algebraic blueprint. This gives us in turn a mechanical interpretation: all configurations of a linkage are considered. That is, the answers provided by ALEPH are mechanically complete. This means that the different components in the algebraic locus provided by ALEPH reflect different configurations in the mechanism giving rise to different mechanical loci.

For example, if we considered the Chebyshev straight line mechanism in Figure 5 and the locus traced by Phun, we see that it is a single closed curve.

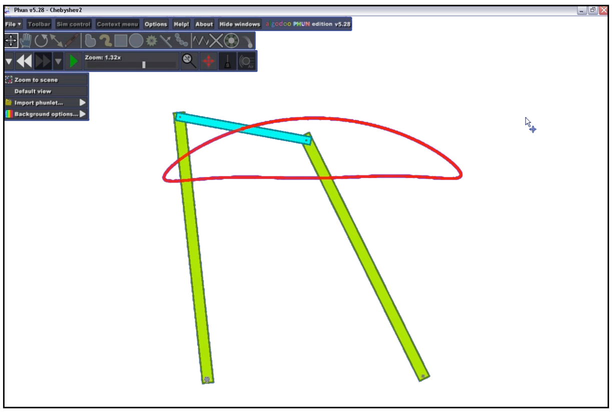

Fig. 5 Chebyshev's locus

However, the graph provided by ALEPH includes two connected componets as shown in Figure 6.

[6] See http://www.mekanizmalar.com/mechanicalspider.html

[7] See http://www.mekanizmalar.com/theo_jansen.html

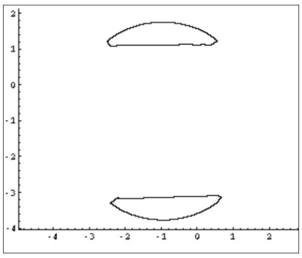

Fig. 6 Graph of Chebyshev's locus given by ALEPH

The second algebraic component corresponds to the trajectory of the same tracer point for the alternative configuration of the same linkage as shown in Figure 7.

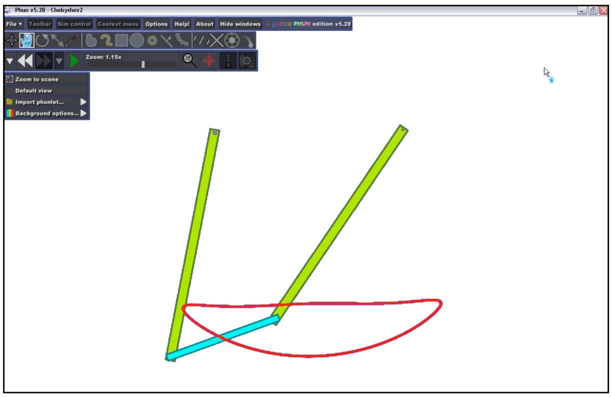

Fig. 7 A different Chebyshev's locus for a different configuration

The linkages that trace a locus curve are those with one degree of freedom. If a linkage has more than one degree of freedom then the locus set traced contains an open set. See for example the locus set traced by a point in a linkage with four bars in Figure 8. ALEPH is able to detect such situations, given the answer The locus is (or is contained in) the whole plane.

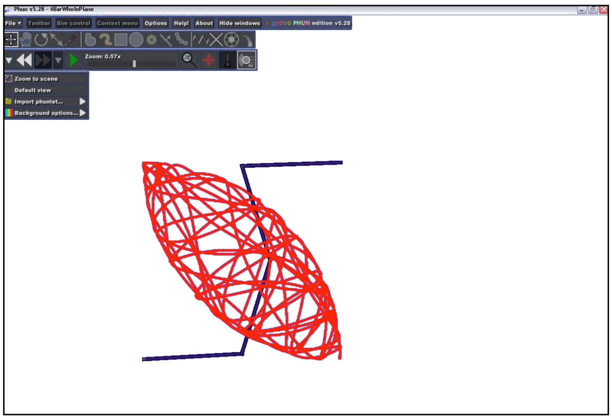

Fig. 8 The locus set is not a curve

2.2 Number of Links and Degree of Equation

The complexity, measured in number of links, of the Kempe linkage associated to a given algebraic curve, has been a topic of research since Kempe first stated his universality theorem. In 2001 Gao et al. proved that for an algebraic plane curve of degree n, a Kempe linkage can be given that uses at most $O(n^4)$ links (Gao et al. 2001). This bound was recently lowered to $O(n^2)$ by Abbot (2008).

Now we are interested in the reciprocal question. Given a proper linkage (i.e. one without useless links) with n links, what is the algebraic degree of the locus traced by a point on it?

The locus traced by the linkage with 5 bars in Figure 9 is part of an algebraic curve of degree 6 as shown in Figure 10.

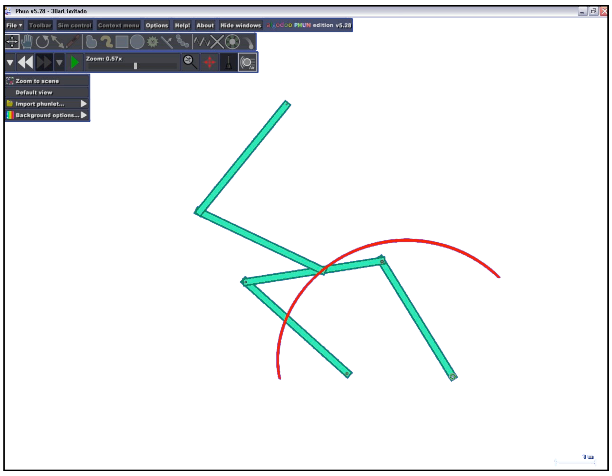

Fig. 9 Curve traced by a linkage with 5 bars

The locus equation is

```
the sextic 5644941897997496389 + 247132437431574000*x - 4244005801580210000*x^2 -
  896317831092000000*x^3 + 780541410100000000*x^4 + 300363840000000000*x^5 +
  28224000000000000*x^6 - 921507192652111800*y - 672376570240320000*x*y +
  352348881966000000*x^2*y + 270401241600000000*x^3*y +
  36667680000000000*x^4*y - 1735122934746650000*y^2 -
  146784399301200000*x*y^2 + 700262042600000000*x^2*y^2 +
  600727680000000000*x^3*y^2 + 84672000000000000*x^4*y^2 -
  579523490100000000*y^3 + 270401241600000000*x*y^3 +
  73335360000000000*x^2*y^3 - 802793675000000000*y^4 +
  300363840000000000*x*y^4 + 84672000000000000*x^2*y^4 +
  36667680000000000*y^5 + 28224000000000000*y^6 == 0
```

Fig. 10 Equation of locus traced by a linkage with 5 bars

In fact, basically the same construction can be done with an arbitrary number of links. This means that we can find a linkage with an arbitrary number of links tracing (part of) an algebraic curve of degree six. This shows in particular that there is no direct relation between the number of links in a linkage and the degree of the curve traced by it.

2.3 Architecture

To obtain the equation of a locus from the corresponding Phun file, a translation to an ad-hoc geometric encoding of the locus in terms of points and point constraints takes place. An example of this was given above in Section 2.

After the aforementioned translating process has taken place, Mathematica (Wolfram 2001) is launched initializing variables according to the specifications in the geometric description of the locus. An initialization file for CoCoA (CoCoATeam 1987) containing the ideal generated by the appropriate defining polynomials is also written out, and CoCoA, launched by Mathematica, computes a Groebner basis for this ideal. Each generator is factored (a task also done by CoCoA), and a process of logical expansion is performed on the conjunction of the generators in order to remove repeated factors. A standard interpretation of the final polynomial generators directly provides the final answer.

The answer given by ALEPH includes the equation of the locus in plain text format allowing the user to re-use the information in other applications.

3 Examples: Four Bar Linkages

In Mechanics, the four bar linkage is the simplest and often times, the most useful mechanism (Hunt 1978). A classical four bar linkage has three moving links and one fixed link (the frame), all joined with four pin joints (hinges in Phun). Their algebraic interest stems from the fact that four bar linkages are the basic elements in Kempe's (flawed but basically correct) constructive proof of the universality theorem (see Kempe 1876; Gao et al. 2001; Abbot 2008). That is, in some sense, Kempe's theorem says that any algebraic curve can be reproduced by (a combination of) four bar linkages.

In Phun, a four bar linkage can be modelled with three bars, two of which are pinned to the background by a hinge (see for example Figures 1 and 5). The following are examples of use of ALEPH for some basic four bar linkages.

More details on these and other examples are available at ALEPH's site[8] where videos demonstrating the movement of all linkages can be found.

3.1 Watt's Linkage

Watt's linkage (also known as the parallel linkage) is a type of mechanical linkage invented by James Watt (1736 - 1819) to constrain the movement of a steam engine piston in a straight line. It can be easily reproduced in Phun with three links (boxes) and three hinges as shown in Figure 1. A pen element has been added to the middle link to trace its trajectory when the linkage is articulated.

---

[8] http://nash.sip.ucm.es/ALEPH

Uploading the .phz file created by Phun to ALEPH the answer provides the graph and equation of the known figure-eight. Figure 11 shows its equation.

The locus equation is

```
the sextic 14689260510002129 - 35446498562391550*x + 13502573647251250*x^2 +
  12399025166500000*x^3 - 2643153325000000*x^4 - 943020000000000*x^5 +
  190125000000000*x^6 - 10945418965759250*y + 11673060293600000*x*y +
  4646411934500000*x^2*y - 3085434300000000*x^3*y + 262470000000000*x^4*y -
  1226072735348750*y^2 + 4013946074500000*x*y^2 - 2087932250000000*x^2*y^2 -
  1886040000000000*x^3*y^2 + 570375000000000*x^4*y^2 + 2131582890500000*y^3 -
  3085434300000000*x*y^3 + 524940000000000*x^2*y^3 + 555221075000000*y^4 -
  943020000000000*x*y^4 + 570375000000000*x^2*y^4 + 262470000000000*y^5 +
  190125000000000*y^6 == 0
```

Fig. 11 Equation of Watt's locus

So Watt's linkage does not generate a true straight line motion. In fact Watt did not claim it did so. Nevertheless it was (and still is) used with that purpose in multiple mechanisms.

3.2 Foldable Crank-Rocker

In a four bar linkage, a crank is a side link which revolves relative to the frame and a rocker is any link which does not revolve. Hence a crank-rocker mechanism is a four-bar linkage in which the shorter side link revolves and the other one rocks (i.e. oscillates).

Figure 12 shows a foldable crank-rocker four bar linkage and the locus traced by a pen on the link in the middle. It is called foldable because it can be moved into a position in which all three bars become simultaneously aligned.

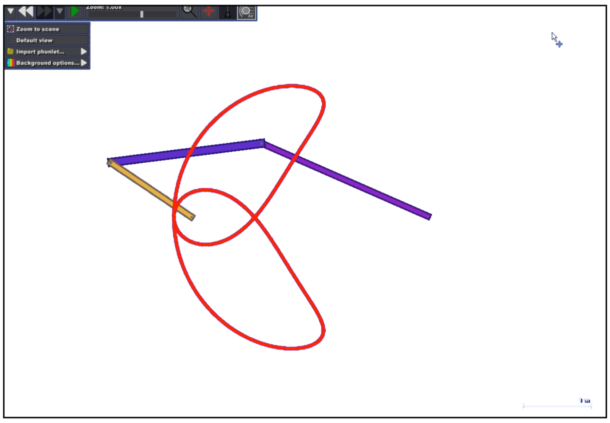

Fig. 12 Locus traced by a foldable crank-rocker four bar linkage

Uploading the .phz file created by Phun to ALEPH the answer provides the equation of the traced trajectory shown in Figure 13.

The locus equation is

```
the sextic 298597991954968 + 277825498535800*x - 15972913206250*x^2 +
  74073751375000*x^3 + 130602358203125*x^4 + 51415234375000*x^5 +
  7222656250000*x^6 + 245404085744250*y + 559655078565000*x*y +
  766748837718750*x^2*y + 357850031250000*x^3*y + 75404531250000*x^4*y +
  1157157571921875*y^2 + 783479479250000*x*y^2 + 482737894531250*x^2*y^2 +
  102830468750000*x^3*y^2 + 21667968750000*x^4*y^2 + 928898274093750*y^3 +
  357850031250000*x*y^3 + 150809062500000*x^2*y^3 + 352135536328125*y^4 +
  51415234375000*x*y^4 + 21667968750000*x^2*y^4 + 75404531250000*y^5 +
  7222656250000*y^6 == 0
```

Fig. 13 Equation of the crank-rocker locus

3.3 Elliptical Trainer

Elliptical trainer is the name given to a popular stationary exercise machine used to simulate walking or running without causing excessive pressure to the joints. The basic frame is a four bar linkage easily reproduced in Phun as shown in Figure 14. A pen element has been added to the middle link to trace the trajectory of a foot when the linkage is articulated.

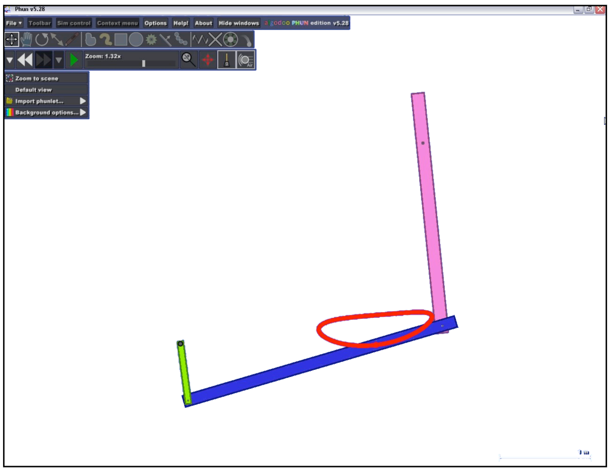

Fig. 14 Locus traced by a foot in an elliptical trainer

Despite the machine's name the trajectory does not seem to be elliptical. Uploading the .phz file created by Phun to ALEPH we get the equation in Figure 15 of degree six!

The locus equation is

```
the sextic 283932031331295361 + 2479000518627766600*x - 11182423829822000O*x^2 -
  1150659616750000000*x^3 - 402293607500000O*x^4 + 117568900000000000*x^5 +
  5041000000000000*x^6 + 722704127337654000*y + 738342427055800000*x*y -
  736146701410000000*x^2*y - 344826836000000000*x^3*y +
  186801000000000000*x^4*y - 192907749755460000*y^2 -
  1383915169830000000*x*y^2 + 204224156500000000*x^2*y^2 +
  235137800000000000*x^3*y^2 + 151230000000000000*x^4*y^2 -
  668951190970000000*y^3 - 344826836000000000*x*y^3 +
  373602000000000000*x^2*y^3 + 244453517250000000*y^4 +
  117568900000000000*x*y^4 + 151230000000000000*x^2*y^4 +
  186801000000000000*y^5 + 5041000000000000*y^6 == 0
```

Fig. 15 Equation of the elliptical trainer locus

4 Conclusions

The work presented here is just an example of the possibilities of intercommunicating Game Physics applications and Computer Algebra Systems, two software paradigms that complement each other.

The leading idea of ALEPH, that is, the concept of a remote add-on, is certainly useful for enhancing the value of other applications.

The tool, although based on hard theoretic mathematical computations has a friendly interface showing also that it is possible to combine the formalism of an academic tool with the current trend of more recreational web services so much in the line of Phun. We believe that this approach opens a wide spectrum of possibilities, from grade school to the engineering student, and even the researcher in automatic proving.

While limited by the computational complexity of some of the algebraic methods involved, ALEPH has been shown to be able to illustrate many non-trivial situations. More involved constructions are expected to be handled by ALEPH once the more computationally efficient Wu's method is implemented as basis for the algebraic computations.

Acknowledgements

This work has been partially funded by the European project Intergeo (ref. IST-507826) and the Spanish project MTM2008-04699-C03-03/MTM

References

Abott, T. G. (2008): Generalizations of Kempe's universality theorem. Thesis (S.M.), Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science. http://hdl.handle.net/1721.1/44375 Accessed 14 June 2010.

Botana, F., & Valcarce, J.L. (2003): A software tool for the investigation of plane loci. Mathematics and Computers in Simulation 61, 139-152.

Buchberger, B. (1985): Groebner Bases: An algorithmic method in polynomial ideal theory. Multidimensional Systems Theory, D. Reidel Publishing Company, 184-231.

Chou, S. (1988): Mechanical Geometry Theorem Proving. Reidel, Dordrecht Boston.

CoCoATeam (1987): CoCoA: a system for doing Computations in Commutative Algebra, Available at http://cocoa.dima.unige.it

Escribano, J., Botana, F., & Abánades, M. A. (2010): Adding Remote Computational Capabilities to Dynamic Geometry Systems. Mathematics and Computers in Simulation 80, 1177-1184.

Gao, X., Zhu, C., Chou, S., & Ge, J. (2001): Automated generation of Kempe linkages for algebraic curves and surfaces, Mechanism and Machine Theory, vol. 36, no. 9, 1019-1033.

Gao, X.S., Zhang, J.Z., & Chou, S.C. (1998): Geometry Expert. Nine Chapters, Taiwan.

Jackiw, N. (2002): The Geometer's Sketchpad v 4.0. Key Curriculum Press, Berkeley.

Hunt, K. H. (1978): Kinematic Geometry of Mechanisms, Oxford University Press, NY.

Kempe, A. B. (1876): On a General Method of describing Plane Curves of the nth degree by

Linkwork. In Proc. of London Mathematical Society, volume 7, 213-216.

King, H. (1999): Planar Linkages and Algebraic Sets, Proceedings of 6th Gokova Geometry-Topology Conference, Turkish J. Math., vol. 23, pp. 3356.

Koebel, A. (2008): Automated Generation of Kempe Linkages for Algebraic Curves in a Dynamic Geometry System. Bachelor Thesis in Computer Science, Saarland University, Germany. http://www.a-kobel.de/kempe/thesis.pdf Accessed 14 June 2010.

Laborde, J. M., & Bellemain, F. (1998): Cabri Geometry II. Texas Instruments, Dallas.

Lebmeir, P., & Richter-Gebert, J. (2007): Recognition of Computationally Constructed Loci. Proc. 6th Int. Workshop on Automated Deduction in Geometry (ADG 2006) LNAI 4869, 52-67.

Richter-Gebert, J., & Kortenkamp, U. (1999): The Interactive Geometry Software Cinderella. Springer, Berlin.

Wolfram, S. (1991): Mathematica: a system for doing mathematics by computer (2nd ed.). Addison Wesley Longman Publishing Co., Inc., Redwood City, CA.