

## TOOLS FOR SYNCHRONOUS DISTANCE TEACHING IN GEOMETRY

**Luiz Carlos Guimarães**

Instituto de Matemática  
Universidade Federal do Rio de Janeiro  
Rio de Janeiro, Brazil  
e-mail: lcg@labma.ufrj.br

**Rafael Barbastefano**

Central de Qualidade  
Fundação Getulio Vargas  
Rio de Janeiro, Brazil  
e-mail: rafael@fgv.br

**Elizabeth Belfort**

Instituto de Matemática  
Universidade Federal do Rio de Janeiro  
Rio de Janeiro, Brazil  
e-mail: beth@dmm.im.ufrj.br

### ABSTRACT

To tackle the problem of teaching Mathematics, using Internet for synchronous communication, our group has developed a suite of three softwares designed to provide an adequate environment. This paper describes some of the characteristics of these tools, and preliminary results of experiments conducted to test how supportive they prove to be in a distance taught course.

The first of these, MathChat, is a general-purpose tool for mathematical communication. With it we can provide tools for

- creating mathematical symbols and equations on the fly,
- symbolic algebra facilities, and
- plotting of functions and surfaces,

this set of tools is integrated with the usual "chat room" facilities.

Tabulae, the second, is a dynamic geometry software with a built in communication server. With it we can instantly set up a virtual classroom, where each student receives, in real time, each step of a geometrical construction the teacher is realizing on his own machine. During the whole process the student is free to modify or add elements of his/her own, and to voice doubts or suggestions. Students can also direct their own work for instant check by the teacher or share it with the "classmates".

Finally there is Mangaba, our 3Dimensional dynamic geometry workhorse. Written in JAVA, it shares most of the communication features available in Tabulae. As an additional feature, it is capable of generating VRML code for any scene the user may construct. The primitives available include a comprehensive repertoire of construction and intersection primitives.

# 1. Introduction

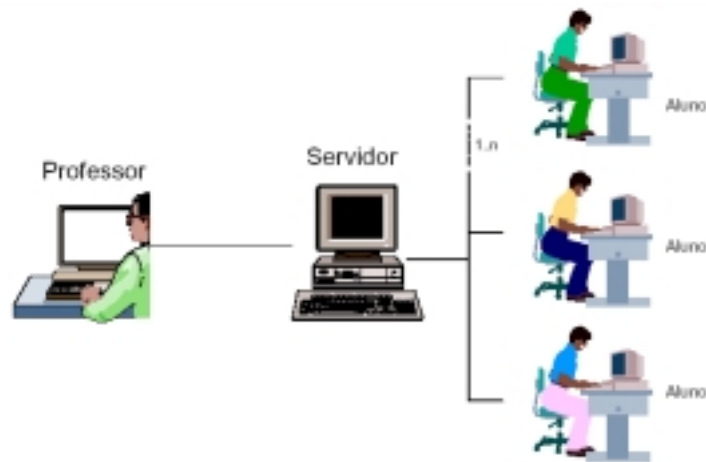
Internet has opened a new dimension for distance education, where synchronous, long distance interaction and information interchange between students and teachers is relatively easy and affordable. Nevertheless, synchronous communication for distance education courses still occurs mostly through Internet Relay Chats, which are essentially tools for the exchange of text messages. This conventional chat room is inadequate for mathematics, where a special language of symbols, diagrams and text was, from very early times, developed to communicate. The combination of these elements is present even in the early texts of Euclid and Apollonius. That mathematical communication and text exceeds the capabilities of the standard tools available in the Internet will be apparent to anyone who browses through some of the many discussion lists specialising in different aspects of the subject.

Of course one might argue that, as a class, mathematicians are able enough to take maximum advantage of the media such as it is, and that we communicate well enough among ourselves, thank you. The long lifespan of some of the discussion lists mentioned above could be evidence of just that. But, much as a master chess player can mentally play several opponents at a time without resorting to a chessboard, it is doubtful that he/she would advocate this as the preferred medium for teaching beginners, especially at a distance.

The ever increasing use of TEX among the community, on the other hand, seems to indicate that, when provided with a more appropriate tool for the job, our average mathematician will prefer not only to read text with the proper symbols, but he/she will take the trouble to set his own text to make it more presentable to the circle of peers. So maybe our master would use a chessboard at home after all.

Diagrams and figures are another matter altogether. Sometimes we are suspicious of them, maybe most times we use them badly [C1] but, when using the blackboard for teaching we so impress engineers that when they try to think up better alternatives for us they come up with oddities such as the “geometer’s workbench” [G]: a panel driven by several computers and several LCD projectors that tries to better us at producing diagrams. So we assume that, at least for teaching, most of us would be only too happy not to have to forsake our special symbols and careful use of diagrams when we are finally driven to use the internet to communicate with our students at a distance.

In doing this we cannot rely on complex apparatus such as a mathematician’s workbench, even it was not unavailable. We must take into account real world limitations and, whenever possible, preserve the formats and tools we already use. For the remaining of this paper we envisage a situation like the scheme depicted in Figure 1, with each student possibly at home and connected to the Internet through a narrow bandwidth telephone modem.



**Figure 1:** A simple scheme for a synchronous lecture.

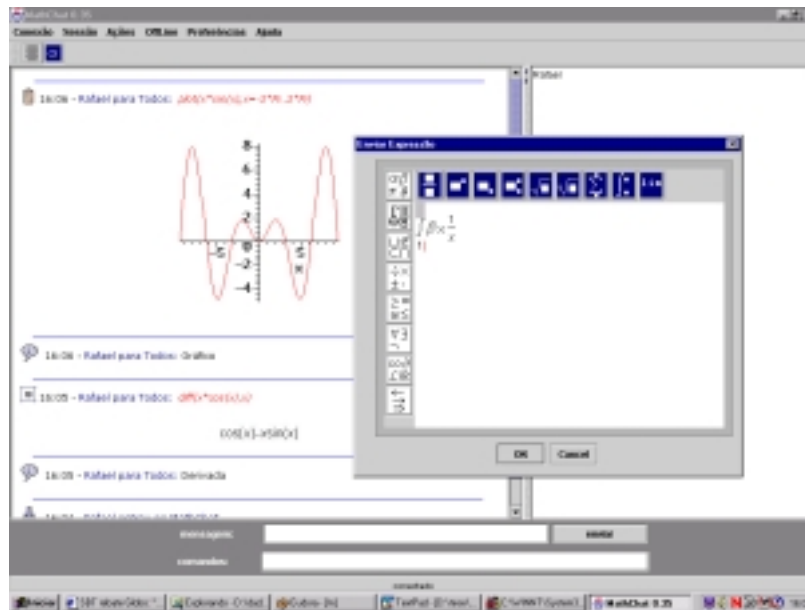
## 2. A double edged approach to formulae and graphics: MathChat

The last ten years have seen an enormous increase in the use of Computer Algebra packages for teaching mathematics, to the extent of developing specialized interfaces to facilitate their use in specific courses and pedagogical applications (see for instance the Metric project, at – <http://www.metric.math.ic.ac.uk>).

Most educational experiments with CAS are intended for the student on campus, at special purpose facilities. Our approach with MathChat, on the other hand, is designed so that the student will have access to the same level of facilities whether on campus, at home or anywhere he/she can connect to the internet. It provides an environment where three complementary means of input are available: a text area similar in most respects to what is available in a “normal” chatroom, an area where command lines can be uploaded to the server, and a formula generation area, where the user can upload formulas and mathematical symbols. The result is displayed to the participants of the chat in a single window, much like a conventional chatroom, except that the messages will include a flexible array of mathematical expressions, graphs, and even animations.

An important constraint for the design brief of MathChat was the demand that all interchanges of information through the network were to be lightweight enough so that sessions are fully interactive, even through low bandwidth networks. That’s why mathematical expressions are transmitted in Latex format, to be rendered locally at each client machine.

MathChat is designed so that it can use a variety CAS as mathematical engines. The current version uses Maple, running on a server machine. In this case the “speaker” uploads a Maple command, or sequence of commands. This is processed at the server, and the final result is transmitted to all participants. If the command generates graphical output, this is sent to the client as a GIF image. Again, mathematical expressions generated by Maple are transmitted to the clients as Latex code, and rendered locally using a version of the WebEq library. See Figure 2 for a screenshot of the work area of the software.



**Figure 2:** A screenshot of MathChat.

An important test bed for this software has been a variety of continuing education courses for high school teachers of mathematics. In this case the lecturer, a fellow mathematician who was also a highly experienced Maple user, would interact with groups of up to fifteen teachers at a time. What surprised us most in the experiment was the intensity of the exchanges. Contrary to what we were led to expect by the literature available on the use of chats for distance teaching, the lecturer could barely cope with the demand generated. The main bottleneck of the system was the viewing area available in the personal computer as the rate of messages grew, and great care was demanded to ensure that no question would go unseen.

As the MathChat project unfolds, we expect the software will be used in a variety of ways. Some groups of users currently running tests to start use in the near future comprehend:

- tutors in distance taught mathematics and physics courses;
- tutors and lecturers in a variety of campus based disciplines, as an aid in the task of helping groups of students with their assignments;
- study groups where students get together at set times in cooperative work centered in different course contents.

### **3. Tabulæ as a tool for distance taught plane geometry**

When the issue of distance teaching is considered, synthetic geometry poses some interesting questions. Consider the examples proposed in [C1] and [C2] as samples of “good” use of figures in mathematical text. Their use is strongly reminiscent of the use of the blackboard by a successful teacher: the drawing is effected step by step, and the student is called in by the instructor to direct attention to the portion of the drawing that is relevant at a given moment. Animations that can be played step by step are an improvement over static figures, and it is quite easy to prepare routines that will do just that on the canvas of a dynamic geometry software. Or they can be written as Java applets, as in [M], where Casselman’s prescription for the presentation of Euclid’s proof of Pythagoras’ theorem is beautifully transposed to an electronic media.

But we wanted an instrument which would give us, for geometry, the same facilities we have with MathChat to perform constructions on the fly. We wanted, in other words, a “blackboard” that would be available at the screen of each student in a lecture delivered live through the Internet, and where the instructor could write each construction step by step, not necessarily according to a previously determined script, but possibly as an impromptu response to a student’s request.

We also wanted more than that: we wanted the student to preserve at all times the complete control of his/her machine, to be able to experiment and propose variations, or even to diddle, in the way he might do while taking notes in a classroom. Therefore a protocol where you surrender control of your machine at the same time you give another person the right to write in it at a distance was out of the question. That precludes the use of general purpose communication tools like NetMeeting.

What we did was to build, into a purpose written dynamic geometry software, an Internet communication server integrated with a “Chat” tool. Its range of uses comprehend:

- as standalone software, for individual use, with the full set of functionalities of a conventional dynamical geometry software;
- a communication vehicle for lectures delivered live through the Internet;
- a vehicle for collaborative and group study, with no constraints posed on relative geographical location of any participants or network bandwidth.

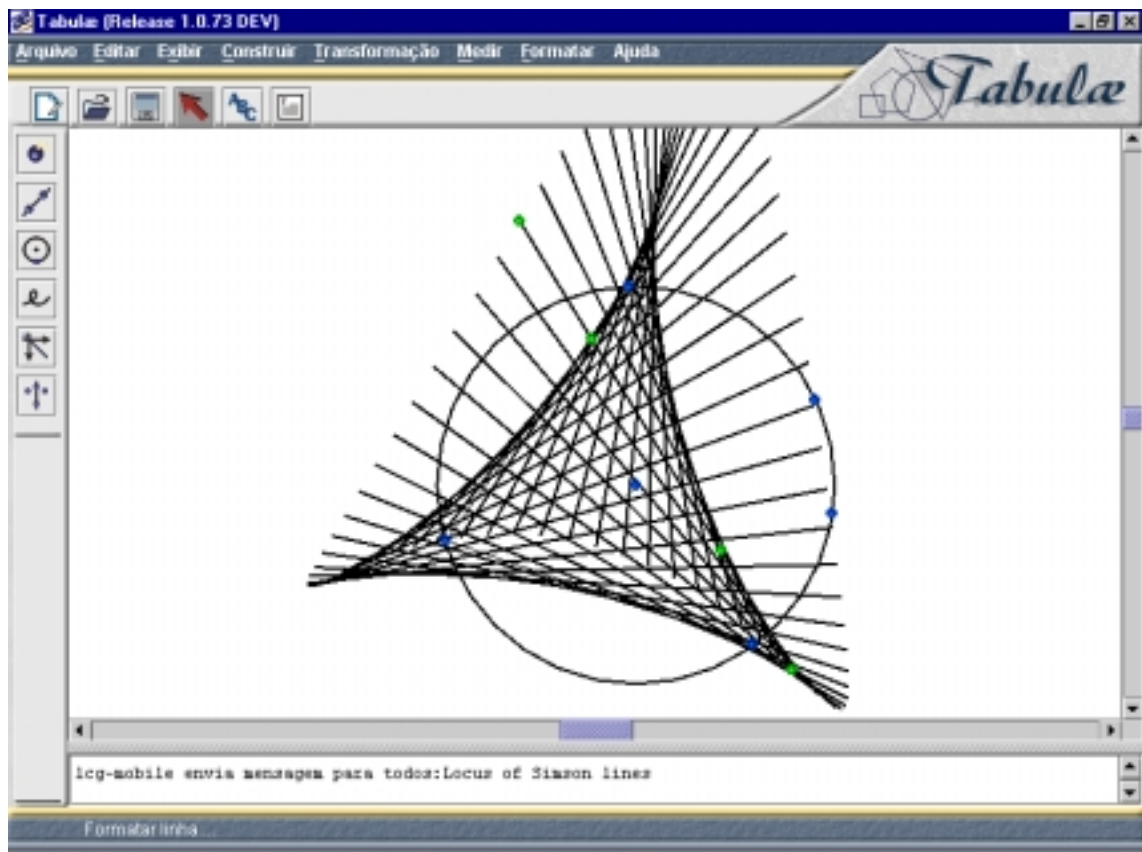
The tool allows a model of Web delivered lecture where, in the same manner as in a traditional geometry classroom, the teacher can gradually construct a geometrical object, while explaining the related content either through an integrated chat window or through a voice channel. A video link is also possible, if the connection bandwidth available to the students permits.

The model puts the teacher in full control not only of each step of construction, but also of the timing to be employed to effect it. Each step is instantly available to every one of the students in the class, who can even experiment with variants of their own while the construction is in progress. The teacher may get feedback from the students through the voice or chat channels, and can even receive a student’s work for exam upon request.

During a problem solving activity a teacher might hand over to a given student the prerogative of transmitting to all his colleagues, for instance. Outside classes students might use the same tool to organize themselves in study groups.

The principle that makes a fast rate of communication viable even through slow networks is that, at no time, we transmit large packets of information. Images, for instance, do not have to be transmitted. Instead, the specifications of a given geometrical object are passed on to the recipient’s copy of the software, which then proceeds to construct an exact equivalent on the local computer. To give only an example: to specify a perpendicular bisector on the student’s machines, the size of the information we need to transmit is under 30 bytes. Even a complex object like a geometrical locus doesn’t use anything significantly larger than that.

The objects constructed by the instructor at the student’s screen are fully functional dynamical geometry constructions, which the student can modify, use as stepping stones for a more complex construction, etc. Of course they can also be saved, as well as the entire session, for later study. Figure 3 shows a screenshot of Tabulæ, including the message line.



**Figure 3:** A screenshot of Tabula. Notice the message line.

## 4. Mangaba: a tool for 3Dimensional geometry

For 3Dimensional geometry, there are no established examples in the market as yet. The conception of a fully functional dynamic geometry software for space geometry is a complex undertaking, and it is only recently that we were able to progress to a point where we can confidently say that we were successful. A full description of the software will be given elsewhere in a forthcoming paper. Figure 4 shows a screenshot of Mangaba.

This is not the place to discuss the difficulties in learning and teaching space geometry. We direct the interested reader to [B], for an account of very basic difficulties of young children, and to [Ch], for a fine analysis of the evolution of the teaching of the subject in France during the last century.

To give an idea of what we are endeavoring to achieve with Mangaba we quote from Hadamard ([H] , vol. II, Book V, pg. 83):

*“In all construction problems we assume, save express indication to the contrary, that we can*

- *construct a line in space given two points;*
- *construct a plane given three points in space;*
- *find the intersection of two given planes, an of a straight line with a given plane;*
- *perform, in any given plane in space, all the known constructions of plane geometry.*

*This assumptions are, of course, merely conventional, and there is no way to perform them in practice. Nevertheless, Descriptive Geometry teaches us to represent, by way of plane*

figures, the figures in space and, in this mode of representation, the constructions referred above can be performed with straight ruler and compass.

We shall call effective constructions those required to be performed with no recourse to the preceding convention.”

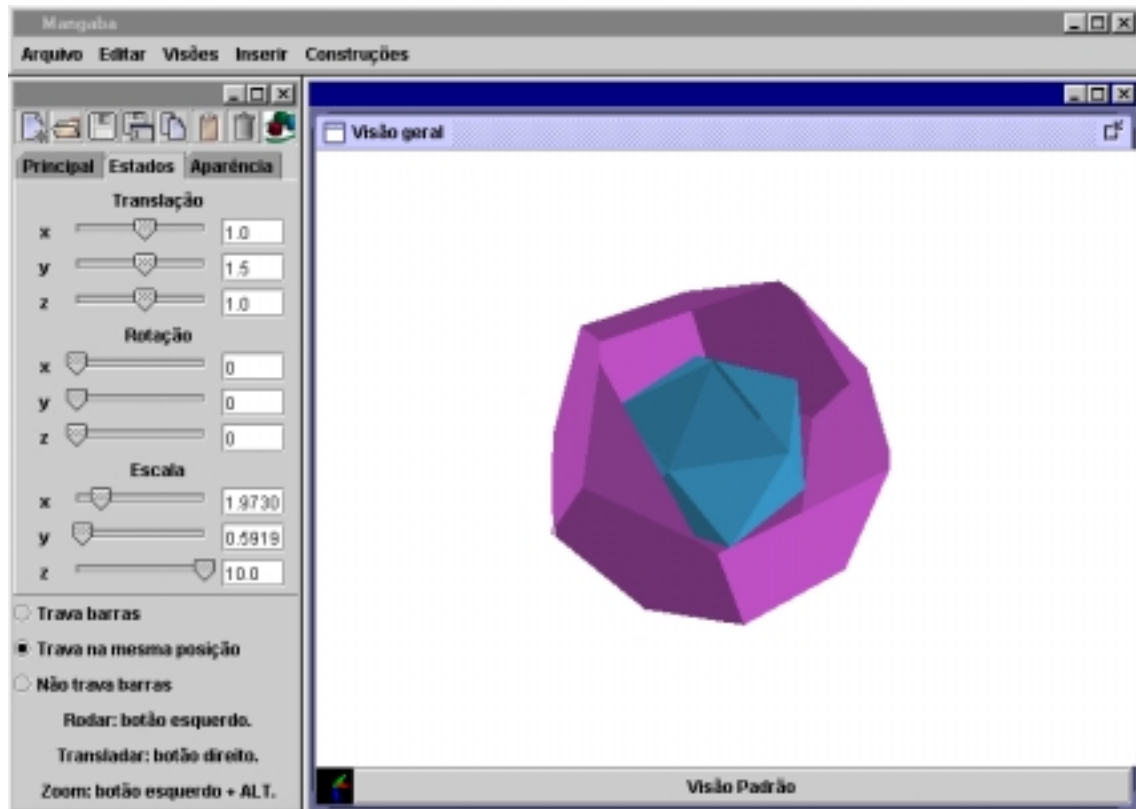


Figure 4: a screenshot of Mangaba.

The excerpt above gives us some indication of the level of abstraction we are dealing with when teaching space geometry, even in simple problems. By way of an example, think of the following exercise, found in any of the classical textbooks of the nineteenth century and, even today, in many textbooks, for instance in France:

*“Suppose you are given two skew lines in space, and a point  $P$  not in either of them. Find a straight line through  $P$  that intercepts both given lines.”*

There are several interesting issues that can be raised by an exercise like that. Some have to do with the status of the diagram depicting the situation, and for that we refer the reader again to [Ch]. But let’s instead think for the moment of the first steps towards solution to the problem: one could think of the data and come up with: the point  $P$ , together with one of the lines, gives us a plane. “Construct” (in the sense of Hadamard) the intersection of this plane with the other line. That, if it’s well defined, gives you a point  $Q$ . The line joining  $P$  and  $Q$  is the sought for solution.

The point here is that the whole operation has to be conducted as a thought experiment. Even after you come to a successful solution, there is very little to show, as the “construction” you achieve is still a thought experiment, and cannot easily be made more concrete. Where you trying to teach this at a distance, there would be very little you could show to help a student come unstuck.

Mangaba deals very easily with simple constructions like this. At the same time, the repertoire of geometrical objects in space geometry, apart from being much larger than what you find in the plane, can include very complex objects. Just think, for instance, of the diverse stellated icosahedra. If we want them in the same footing as text in the available vocabulary for a synchronous interchange through the Internet, we clearly have to specify with care our exchange dialect. The procedure we have adopted is a little different from the one we use with Tabulæ, but it allows us to specify a complex object, or a geometrical transformation performed on it in the student's machines, with very little more expense in terms of bandwidth than for the case of the simplest object, i.e. the point.

**Acknowledgements:** The project discussed in this article was partially financed by CNPq.

#### REFERENCES

- [B] Battista, M. T., Clemens, D.H., 1998. Student's understanding of Three-Dimensional Cube Arrays: Findings From a Research and Curriculum Development Project. In R. Lehrer & D. Chazan (eds:) *Designing Learning Environments for Developing Understanding of Geometry and Space*, pps. 227-248. Mahwah: LEA.
- [C1] Casselman, B., 2000. Pictures and Proofs. In *Notices of the AMS*, November, Volume 47, Number 10, pp. 1257-1266. Providence: MAS.
- [C2] Casselman, 1999. Visual Explanations. In *Notices of the AMS*, January, Volume 46, Number 1, pp. 43-46. Providence: MAS.
- [Ch] Chaachoua, A., 1997. *Fonctions du Dessin dans l'Enseignement de la Géométrie dans l'Espace*. Unpublished PhD. Thesis. Grenoble: LEIBNITZ-IMAG.
- [G] Grimbetière, F., Winograd, T., Wei, S. X., 2001. *The Geometer's Workbench: An Experiment in Interaction With a Large, High Resolution Display*. Stanford: Computer Science Department.
- [H] Hadammard, J., 1988 (repr.). *Leçons de Géométrie*. Paris: Gabay.
- [M] Morey, J., 1997. *Pythagoras' Haven*. Java Applet at: [www.math.ubc.ca/~morey/java/pyth/1index.html](http://www.math.ubc.ca/~morey/java/pyth/1index.html).