

Computer-aided strategies to shorten the gap between high school and university math courses

A Morante and José A Vallejo

amorante@fciencias.uaslp.mx, jvallejo@fc.uaslp.mx

Facultad de Ciencias

Universidad Autónoma de San Luis Potosí

SLP 78290

México

Abstract

We present some selected topics from an introductory (first-year) course in Mathematics for engineering and science students. The course covers standard topics and techniques of pre-calculus, but with two distinct features: first, we make heavy use of free mathematical software, in particular the CAS Maxima and the DGS Geogebra; second, the examples illustrating the theory are drawn from real-life situations, using actual data obtained from the internet. In this paper we show some samples of computer lab practices, related to the use of entropy in genomics, reflection properties of Cassegrain antennas, and the Reed–Solomon correcting code method.

1 Introduction

Science and engineering students entering the university, frequently meet a disappointing situation whose origin lies in the gap between the mathematics they learnt in high school, and the more abstract taught in calculus or algebra courses in their first year. But it is not only that they can not grasp the intricacies of a proof or the subtleties of a definition. Very often they get stuck at simple computational problems, or can not make the first steps to model a physical situation, so they can not put it in terms that can be solved algorithmically.

The straightforward consequence is that they get low grades in math, even though they got “good” ones in high school, and at our faculty, about one third of them quits after the first year. This is a common and serious problem in many universities in Mexico, and surely exists in other developing countries.

The gap between a high school education, based on memorizing formulas, routine calculations and artificial problems solving, and the math skills required for the more advanced courses, including formal definitions of abstract concepts and simple proofs, is insurmountable. Just as an example, we show the statistics of the grades of 330 new students that got a place at our faculty in 2014. The final

grade (calif_admn) is the weighted mean of a psychometric test (ex_psic), a specific science test¹ (fc) and a general skills test EXANI-II² (ceneval).

Science test (fc)			
Min	23.00	1st Qu	37.00
Median	43.00	Mean	44.11
3rd Qu	50.00	Max	78.00

The median score in the science test (which is a five-options one, with no discount for fails) is 43/100, being located 30/100 points below the general skills test and 20/100 points below the psychometric test.

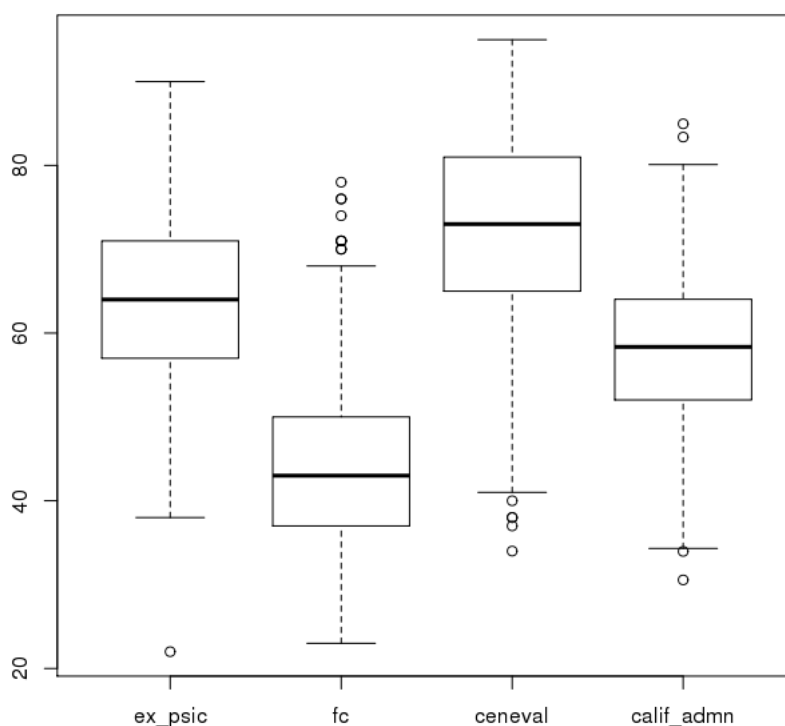


Figure 1: Grades for the access exam in Science Faculty UASLP (2014).

This weakness in scientific knowledge, particularly in Mathematics, asks for a solution requiring a wide range of strategies, most of them out of our reach as university teachers (such as the one related to the curricula in high schools). Nevertheless, in order to offer motivation and trying to upgrade the

¹Topics covered in this exam are Mathematics, Physics, Chemistry and Biology, all of them at a very basic level.

²This is a standardized test for entering colleges in Mexico.

students performance, we have designed a hands-on math course based on the use of technology and a problem-solving approach (true real-life problems, using data accessible in internet). The course makes a strong use of *free* mathematical software, in particular, the computer algebra system (CAS) Maxima and the dynamical geometry software (DGS) GeoGebra. The structure of each lesson is the same: first, some basic notions, presented in an informal way and illustrated with numerous “paper and pencil” examples. Then, an application to some problem which involves more advanced ideas (frequently drawn from other fields such as Biology or Electric Engineering) but requiring the previous basic notions for their solution. After completing the applications section, our students are motivated enough to be assigned a set of homework problems. Next follows a computer lab where the student learns how to use mathematical software to efficiently carry on the computations done “by hand” up until now. Finally, some more homework, this time to be done with the aid of the computer, is presented. Here is a table showing the lessons contents, the applications and the software used.

Lesson topics	Applications	Software
Elementary Arithmetic	Golden ratio, Marquardt mask	Maxima
Modular Arithmetic and divisibility	The ISBN system	Maxima
Prime Numbers. LCM and GCF	RSA cryptography	Maxima
Induction and binomial coefficients	Counting and Probability	Maxima
Maps and Functions	Physics of sports	Maxima, GeoGebra
Exp, Log and absolute value	Information theory and entropy in Biology	Maxima, GeoGebra
Trigonometry	Basic Astronomy. Dead reckoning navigation	GeoGebra, Maxima
Analitic Geometry	How to make a slot for a chimney in your roof. Cassegrain antennas	GeoGebra, Maxima
Complex Numbers	Electric circuit theory, phasors	GeoGebra, Maxima
Polynomials and Roots	Braking distances. Reed–Solomon codes	Maxima, GeoGebra

In the next three sections, we show some sample computer laboratories developing some of these topics. The last section is devoted to some conclusions that we have reached after teaching this course.

2 The log function: Entropy and genomics

Consider a source of information F (a newspaper, a podcast, a computer program, a DNA string, etc.). This source can construct its information out of n distinct characters, called the alphabet of the source (26 letters in the case of the newspaper or the podcast, the symbols 0 and 1 in the case of the

computer program, or the characters A, C, G, T for the DNA string). We call a code of length k any aleatory sequence of k characters taken from the alphabet, allowing for repetitions.

Notice that this definition forgets about the semantic content: a code such as *'akenrbfyde'* is as valid as *'table'*. Also, a basic feature of the codes is that the more unexpected they are, the greater information they provide. For instance, a patient gets much more information from the code *'you tested negative'* than from *'the results are inconclusive'*, despite the second being longer.

When dealing with information sources, we must distinguish between two cases. In the first one, we only know the total number N of codes that can be emitted, without additional details about each individual code. In the second one, we can assign to each code a relative weight (for instance, the probability it appears).

► Equiprobable codes

In this case, it is easy to define a measure of the global information provided by the source F . If it can emit N codes of length k , we put $\tilde{I}(F) = N$. This measure has several advantages: is computable, positive, and it grows linearly with N , as one would expect. However, it does not have the additive property. We want the total information of two different, unrelated, sources $I(F_1 + F_2)$, to be the sum of the partial informations, $I(F_1) + I(F_2)$. But if F_1 can emit N_1 codes of length k_1 , and F_2 can emit N_2 codes of length k_2 , taking them together to emit codes of length $k_1 + k_2$ (for instance, joining the respective codes), we get $N_1 \cdot N_2$ codes, not $N_1 + N_2$. Of course, logarithms provide the solution to this problem. If, instead of $\tilde{I}(F)$, we use

$$I(F) = \log(N), \tag{1}$$

as the global information measure, we get almost all the desirable properties: as $N \geq 1$ (otherwise, there would be no codes), it is $I(F) \geq 0$. Moreover, by the properties of \log , if the codes from F_1 and F_2 are all distinct, $I(F_1 + F_2) = I(F_1) + I(F_2)$. However, in general there will be codes that can be emitted by either source, so we get subadditivity

$$I(F_1 + F_2) \leq I(F_1) + I(F_2).$$

The only lacking property is normalization. We would expect the minimal, non trivial, information provided by the source to have measure 1, but this does not happen with our definition (1), because we do not get the minimal info when $N = 1$ ³. The minimal information is achieved with $N = 2$, so suppose we have a source that can emit two different codes, both with the same probability. Then, definition (1) gives $I(F) = \log(2)$. In order to get the value 1, we must take the logarithm in base 2, and this is the meaning of the normalization: to fix the logarithm base. Now, we can define the normalized information measure of the source F as

$$I(F) = \log_2(N).$$

In this case the unit of information is a bit (binary digit), $1 \text{ bit} = \log_2(2)$. For instance, if we take as the source F the tossing of a coin, the possible codes of length 1 are H and T (head and tail), each one appearing with a probability $1/2$. Then we have $N = 2$ and the normalized information measure for this process is

$$I(F) = \log_2(2) = 1,$$

³As in this case, there is only one message that can be emitted, so there is no new knowledge: we already know what the source will say

that is, one bit of information. If the process consists in tossing the coin k times, we have codes of length k , and the source can emit 2^k of them. Thus, the information gained with each tossing is now

$$I(F) = \log_2(2^k) = k,$$

as expected.

► **Weighted codes**

This time, suppose a source of information which can emit N distinct codes m_i (with $i \in \{1, 2, \dots, N\}$), each one with a different probability of appearance, that we will call p_i . These must satisfy

$$\sum_{i=1}^N p_i = 1.$$

On the other hand, notice that whenever $0 < p_i < 1$, it is $\log_2(p_i) < 0$, so if the p_i represent probabilities, $-\log_2(p_i) > 0$ ⁴.

Now we come to a basic idea in information theory: each code can be considered as a microsource in itself, whose information measure is given precisely by the probability it is emitted

$$I(m_i) = -\log_2(p_i), \quad i \in \{1, 2, \dots, N\}.$$

A motivation for introducing this apparent complication, is that, whenever p_i approaches zero (in any logarithm base), its associated information $I(p_i)$ grows indefinitely. That is, in this way we give a precise mathematical sense to the aforementioned feature of the codes, that the information they provide is greater when they are unexpected.

Therefore, in the case of weighted codes we can define the global information measure of the source as the weighted average of the information provided by the microsources that conform it:

$$I(F) = \frac{\sum_{i=1}^N p_i \cdot I(m_i)}{\sum_{i=1}^N p_i},$$

or, taking into account the restriction $\sum_{i=1}^N p_i = 1$,

$$I(F) = -\sum_{i=1}^N p_i \log_2(p_i).$$

The information measure expressed in this way (for the case of different codes having different probabilities of being emitted), is called entropy, from the greek $\tau\rho\acute{o}\pi\omicron\varsigma$ (*tropos*, which means ‘rotating’). Intuitively, one speaks of the entropy as a measure of disorder: in a disordered room, for instance, knowing the location of a certain object gives much more information than knowing it when everything is in its place.

⁴If a code had $p_i = 0$, it could not be emitted, so we can take $0 < p_i$. And, in case any of them had $p_i = 1$, that code would be the only one emitted, and all the discussion would be senseless. Thus, we can also take $p_i < 1$.

Let us see an application of these notions in Biology. The following sequence is a DNA portion corresponding to the bacterium E. Coli (see, for instance <http://regulondb.ccg.unam.mx/>. There are dozens of on-line databases related to E. Coli.):

AGCTTTTCATTCTGACTGCAACGGGCAATATGTC

It is well known that proteins are synthesized from DNA; concretely, each block composed of three bases among A the (adenosin), C (cytosine), G (guanine) and T (thymine), called a ‘codon’, encodes an amino acid, which are the primary constituents of proteins; for instance, CAG is the codon for glutamine. We can estimate the appearance probability of each codon by a direct counting in our DNA fragment:

AAT	1	AAC	1	ACG	1	ACT	1	AGC	1
ATA	1	ATG	1	ATT	1	CAA	2	CAT	1
CGG	1	CTG	2	CTT	1	GAC	1	GCA	2
GCT	1	GGC	1	GGG	1	GTC	1	TAT	1
TCA	1	TCT	1	TGA	1	TGC	1	TGT	1
TTC	2	TTT	2						

Thus, the probability for the codon AAT is $1/32$, while for TTC we find $2/32$ (notice that, in the case of TTT, there are two instances, because the segment TTTT contains it twice: TTTT and TTTT). The entropy for this particular DNA fragment is:

$$-\sum_{i=1}^{32} p_i \log_2(p_i) = -\left(5 \cdot \frac{2}{32} \log_2\left(\frac{2}{32}\right) + 22 \cdot \frac{1}{32} \log_2\left(\frac{1}{32}\right)\right) = 4.6875 \text{ bits}.$$

Of course, deepening in this topic is far beyond the reach of these notes (for more info, see e.g. [6, 2]), but at least we can remark that this kind of analysis is very useful in Genomics. For instance, it is very easy to prove –using calculus– that the maximum entropy for a given fragment is attained when all the codons are equiprobable⁵ and, then, its value is $\log_2(27) = 4.755$ bits. Thus, we can conclude that this particular fragment of DNA contains a lot of information from the point of view of Genetics, as its entropy is relatively near the maximum value.

Due to the length of a typical relevant DNA fragment (containing thousand of codons⁶), the paper-and-pencil analysis carried so far is no longer feasible, and the use of a computer is unavoidable. In the computer lab corresponding to this topic, the students are asked to write a Maxima script that computes the entropy of DNA sequences downloaded from the database <http://regulondb.ccg.unam.mx/>, and then to compare it for different proteins.

⁵Let \mathbf{p} be the probability vector of all the codes, $\mathbf{p} = (p_1, \dots, p_N)$ with p_i the probability of code m_i . Let us see that $I(\mathbf{p}) = -\sum_i p_i \log_2(p_i)$ reaches a maximum precisely when $p_i = 1/N$ for any $i \in \{1, \dots, N\}$. Changing from base 2 to base N in the logarithm and imposing that the partial derivative of $I(\mathbf{p})$ with respect to each p_j vanishes, we get:

$$\frac{\partial I}{\partial p_j} = \frac{-1}{\log_N(2)} (\log_N(p_j) + 1) = 0,$$

and this happens if and only if $-\log_N(p_j) = 1$, that is, $p_j = 1/N$.

⁶The total number of letters in the E. Coli DNA code is above four million.

3 Analytic Geometry: Cassegrain antennas

The parabola is the set of points in the plane that are equidistant from a fixed line (the directrix), and a fixed point (the focus) lying outside the line. It is well-known that it has the following reflection property: when a beam of light, parallel to the axis of symmetry, incides on a parabola, the rays are reflected in a way that all of them pass through the focus. This property can be studied with the aid of GeoGebra, with the following construction:

- Crete a slider p , setting the minimum and maximum values as $-5, 5$, and the initial value to, say, $p = 2$.
- Write the standard equation for the parabola with vertex the origin and focus $(0, p)$.
- Graphically determine the focus, and rename it as F .
- Select a point on the parabola, and rename it as P .
- To simulate a light beam, or a sound wave emanating from the focus F , draw the straight line FP , and rename it as l .
- Draw the tangent line to the parabola through the point P , and rename it as t .
- Draw the normal line to the parabola through P , and rename it as n .
- Draw the image of the line l reflected on n .

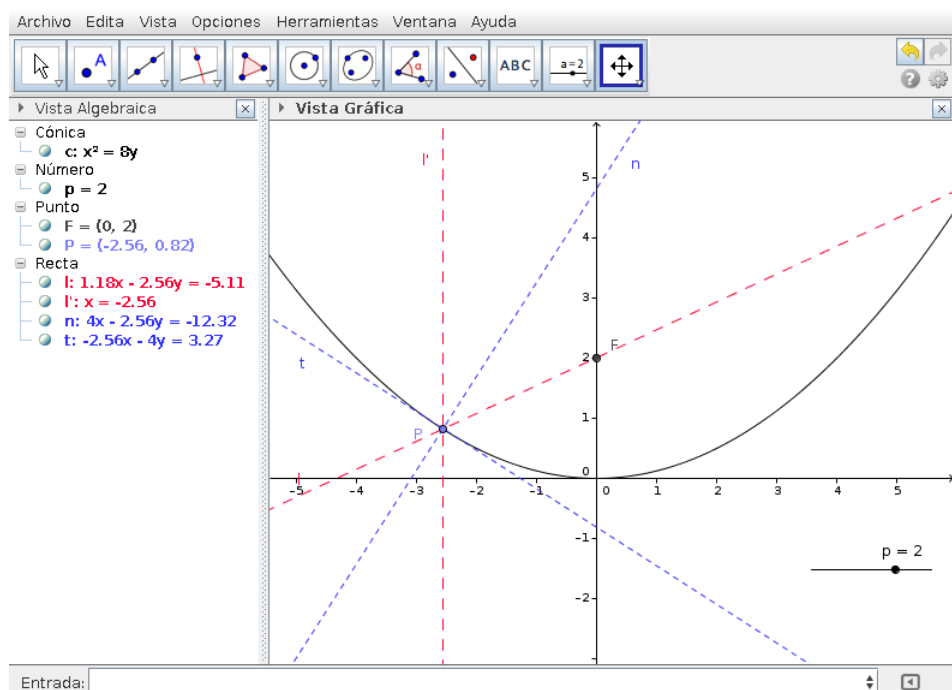


Figure 2: Reflection properties of a parabola.

Similarly, a hyperbola is the set of points in the plane the difference of whose distances from two fixed distinct points (the foci) is a given positive constant. It has the following reflection property: a beam of light directed at one of the foci will be reflected by the curve through the other focus. This is the GeoCebra construction that allows us to study this fact:

- Create two sliders a and b with positive values, say, with 1 as a minimum and 5 as a maximum, and set their initial values as $a = 3$, $b = 2$, respectively.
- Write the standard equation for the hyperbola whose midpoints of its central rectangle are $(a, 0)$, $(-a, 0)$, $(0, b)$ and $(0, -b)$ (see, for instance, [5] pg. 785–787, or the nice exposition in [4]).
- Determine the foci of the hyperbola, and rename them as F_1 and F_2 .
- The hyperbola divides the plane into three regions; select a point O in the one that does not contain any of the foci.
- To simulate a light beam, or a sound wave emanating from the point O directed towards F_1 , draw the straight line F_1P , and rename it as l .
- Determine the intersect of l with the branch of the hyperbola containing F_1 , and rename it as P .
- Draw the tangent to the hyperbola through the point P , and rename it as t .
- Draw the normal to the hyperbola through the point P , and rename it as n .
- Draw the image of the line l and the point O reflected on n .
- Draw the vectors from O to P , and from P to O' , renaming them as u and u' , respectively.

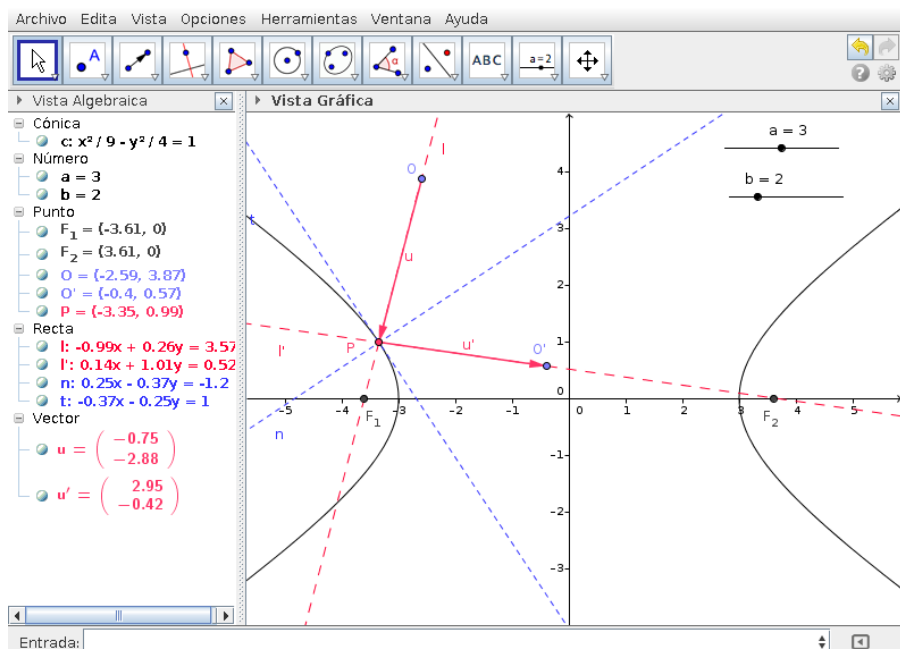


Figure 3: Reflection properties of a hyperbola.

Let us see how these properties are applied in real life. In order to reduce the size of parabolic antennas without power loss (among other benefits, see [1, 3]), the so-called Cassegrain antennas are used: they are formed by a parabolic primary reflector that shares its focus with a hyperbolic secondary reflector.

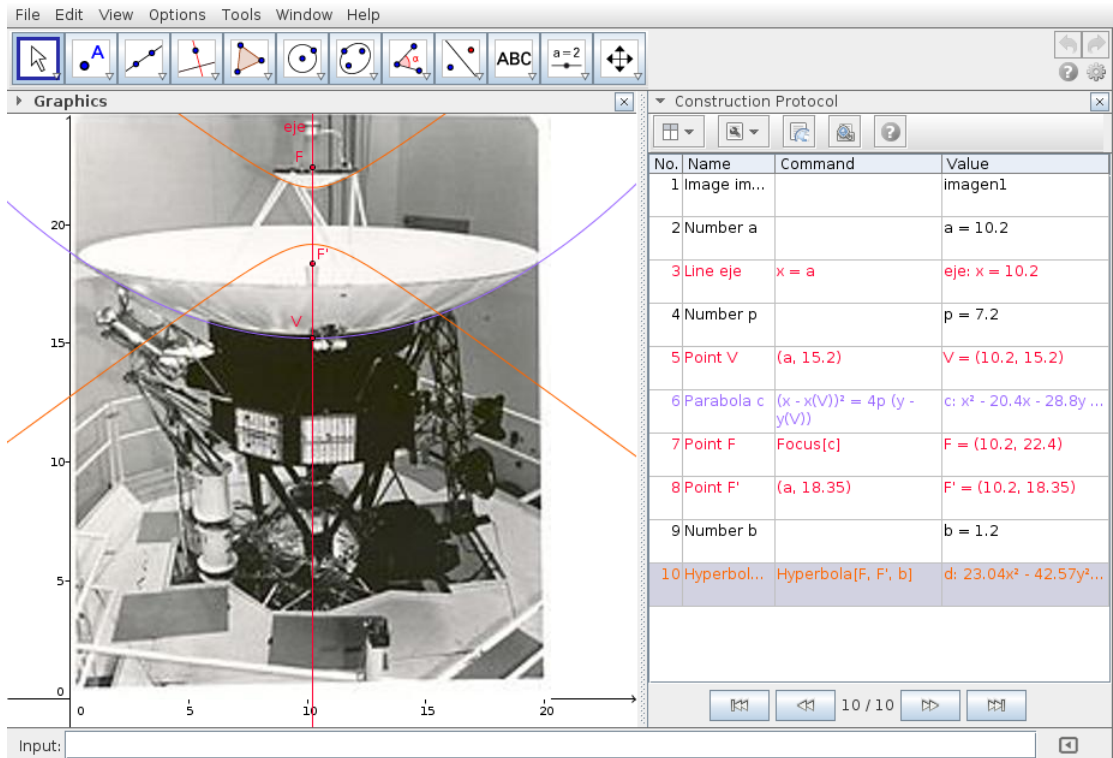


Figure 4: Voyager antenna.

Figure 4 shows the Cassegrain Voyager antenna, with a trial and error superimposed parabola and hyperbola on the left side pane of the image (with the unavoidable perspective error induced by the photograph quality). The photograph used here can be download from the NASA repository, at the URL <http://dayton.hq.nasa.gov/IMAGES/LARGE/GPN-2003-000008.jpg>.

Following the steps of the Geogebra construction showed in Figure 5, we can verify that a vertical beam falling into the primary reflector (in this case represented by the vector u), goes trough the hyperbola focus (in this case at $F' = (10.2, 18.35)$). Our students are asked to prove this fact in the computer lab corresponding to this topic.

4 Polynomials: Reed–Solomon corrector method

For a given set of n different points $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, is possible fit polynomials of degree k for each k between 1 and $n - 1$. In particular, if the abscissas of the points are all different, say $x_1 < x_2 < \dots < x_n$, the polynomial of degree $n - 1$ will be the interpolating polynomial $p(x) = a_{n-1}x^{n-1} + \dots + a_1x + a_0$, passing through each one of the given points, that is, $p(x_j) = y_j$

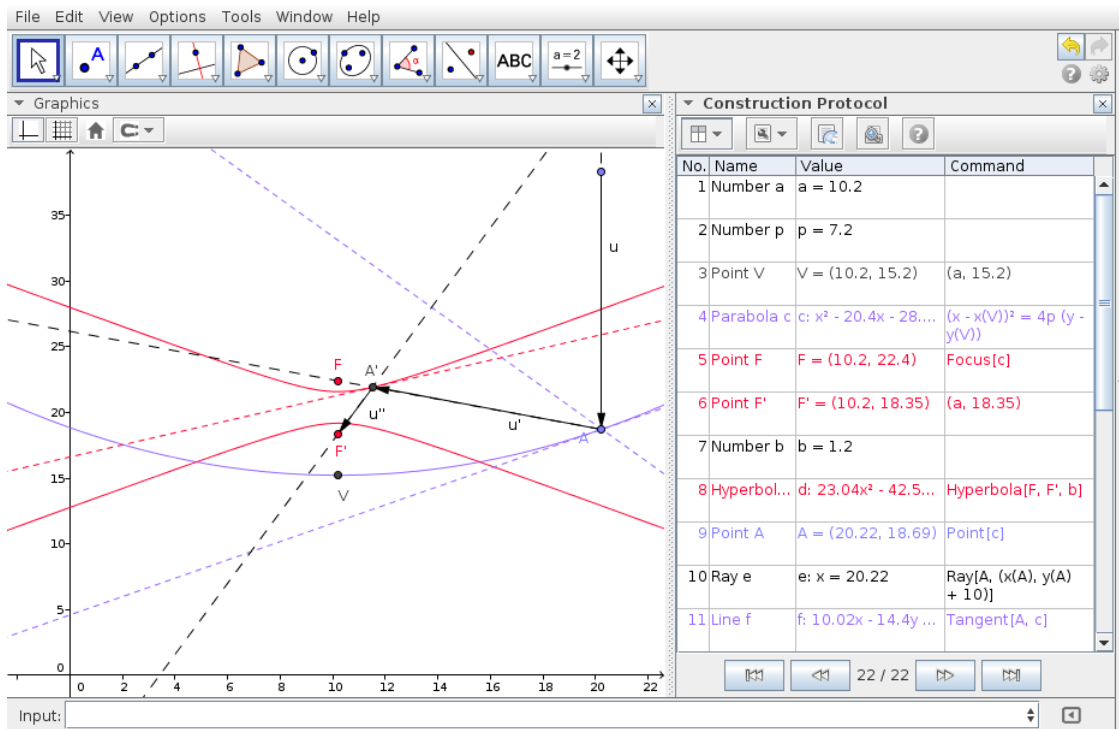


Figure 5: Reflection properties of Cassegrain antennas.

for each $j = 1, 2, \dots, n$. In order to interpolate n different points, we can use any of the following equivalent commands in GeoGebra:

- `FitPoly[List of Points, n-1]`
- `Polynomial[List of Points]`

Interpolating polynomials can be used to fix reading errors when coded data are sent through a digital channel (for example, digital photos or CD audio). The Reed–Solomon method is widely used to fix broken codes (see a list of applications in the Wikipedia page [https://en.wikipedia.org/wiki/Reed–Solomon_error_correction#Applications](https://en.wikipedia.org/wiki/Reed%E2%80%93Solomon_error_correction#Applications) or the detailed book [7]), and next, we show how it works. Suppose we want to transmit the numbers

$$1.2, -3.2, -5.4, -1.1,$$

over a noisy channel. To keep the integrity of the code we add some redundant data to the original values; in this example these data will be a sequence of the same length that the original data sequence (the number of additional data depends on the actual transmission system error probability, with more data required for higher probabilities). Interpolating the corresponding time series (see Figure 6):

$$O_1 = (1, 1.2), O_2 = (2, -3.2), O_3 = (3, -5.4), O_4 = (4, -1.1),$$

we get the interpolating Lagrange polynomial,

$$p(x) = 0.717x^3 - 3.2x^2 + 0.183x + 3.5.$$

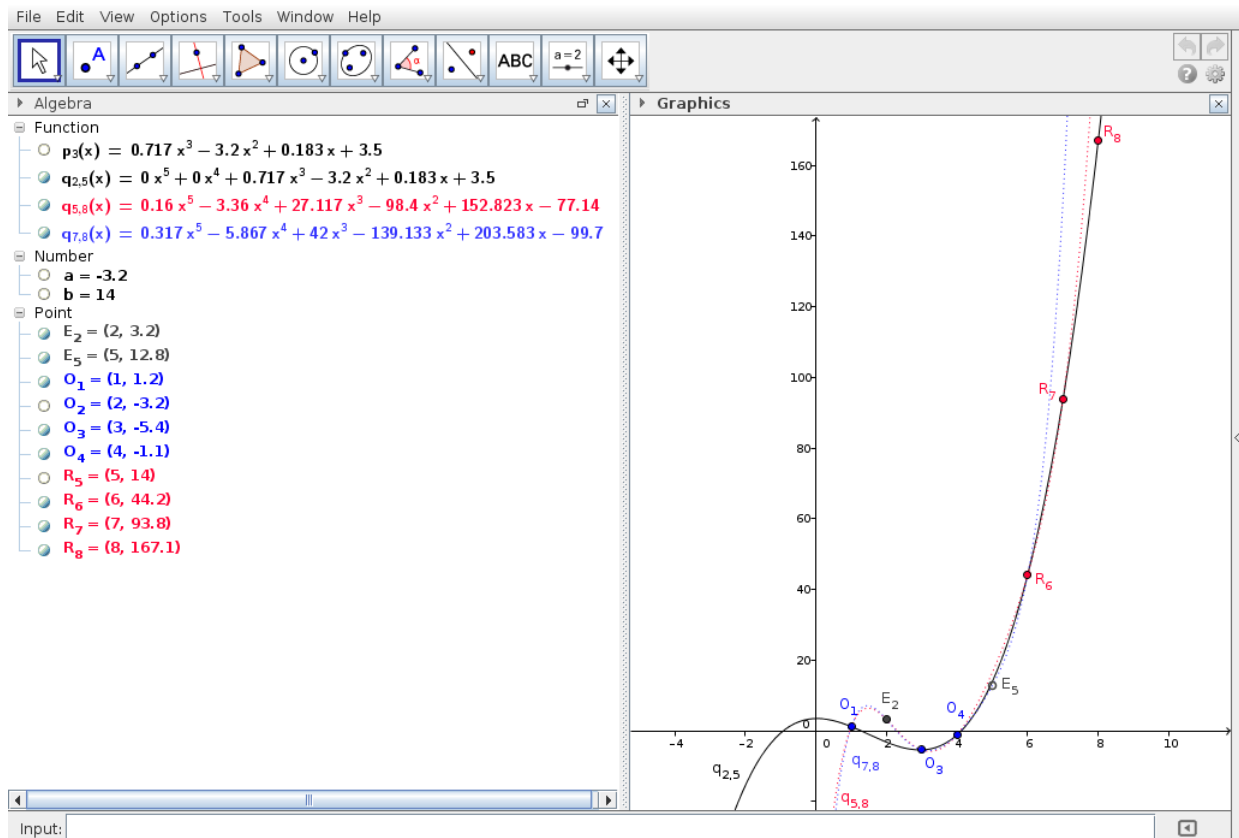


Figure 6: Interpolating polynomial to fix broken data.

We extrapolate the values $p(5), p(6), p(7), p(8)$, to get the extended sequence:

$$1.2, -3.2, -5.4, -1.1, 14, 44.2, 93.8, 167.1.$$

These eight numbers are transmitted keeping their order; let us suppose that the proportion of broken data is at most 25% of the total sent (in this example we suppose that these are the second and fifth values):

$$1.2, \mathbf{3.2}, -5.4, -1.1, \mathbf{12.8}, 44.2, 93.8, 167.1.$$

Since we are doubling the number of values sent, the decoder can determine that we are using a cubic interpolating polynomial. Now, if the decoder interpolates the full time series

$$O_1 = (1, 1.2), E_2 = (2, 3.2), O_3 = (3, -5.4), O_4 = (4, -1.1), \\ E_5 = (5, 12.8), R_6 = (6, 44.2), R_7 = (7, 93.8), R_8 = (8, 167.1),$$

including the broken data, it is possible to detect if there exists errors: if the interpolating polynomial is of degree 3, there are no errors. If this is not the case, we omit each of the received points, one at a time: if omitting the i -th point results in a cubic polynomial, say $q_i(x)$, the i -th value is the only one broken, and its correct value is $q_i(i)$. Following this approach, in our example we omit pairs of received points, one pair at a time: when the second and fifth points are omitted, we get the cubic

interpolating polynomial $q_{2,5}(x) = 0.717x^3 - 3.2x^2 + 0.183x + 3.5$ and, therefore, the only broken data are the second and the fifth, with corrected values: $q_{2,5}(2) = -3.2$ and $q_{2,5}(5) = 14$.

In the computer lab, we ask our students to write a script in Maxima to analyze a given code, correcting its errors (if any) automating the steps above.

5 Conclusions

The gap between the math skills developed in high school and those required in college instruction is a common and serious problem in many universities, that results in low grades and, eventually, the dropout of science and engineering students entering the university. Although this situation requires deep changes in the high school curricula, most of them out of our reach as university teachers, we feel that it is possible to shorten this gap through introductory, non-traditional courses, oriented to ameliorate the engagement and performance of new students.

With this goal in mind, we have designed and taught at the Science Faculty of our university (UASLP), an introductory first-year pre-calculus course, based on a hands-on approach using mathematical software (the CAS Maxima and the DGS Geogebra) and the resolution of real-life situations, using actual data obtained from the Internet. This is illustrated in this paper via selected computer lab practices.

Even though a serious statistical analysis to check if there exists a performance improvement for students having taken this introductory course has not been performed (the sample would be quite small), the grades these students got in the subsequent one variable calculus and linear algebra courses seem to indicate there were some improvements. This is so particularly in the case of the students who, even failing in the overall grade, did better in the access exam (it should be remarked that it is possible for a student to fail in the access exam and to enter the faculty, because of the low demand). This could be taken as a sign that the course caught their interest. Also, a survey made after the course finished, reflects that at least the students enjoyed the use of mathematical software and the down-to-earth examples introduced.

References

- [1] S. Drabowitch, A. Papiernik, J. Encinas, H. Griffiths, B. Smith: *Modern Antennas*. Chapman and Hall (1998).
- [2] L. Frappat, C. Minichini, A. Sciarrino, P. Sorba: *Universality and Shannon entropy of codon usage*. Phys. Rev. **E68** (2003) 061910.
- [3] W. A. Imbriale: *Large Antennas of the Deep Space Network*. Wiley and Sons (2003).
- [4] K. Kendig: *Conics*. Mathematical Association of America (2005).
- [5] R. Larson: *Elementary and Intermediate Algebra*. Brooks/Cole (2009).
- [6] R. N. Mantegna, S. V. Buldyrev, A. L. Goldberger, S. Havlin, C. K. Peng, M. Simons, H. E. Stanley: *Systematic analysis of coding and noncoding DNA sequences using methods of statistical linguistics*. Phys. Rev. **E52** (1995) 2939.

- [7] S. B. Wicker, V. K. Bhargava: *Reed-Solomon Codes and Their Applications*. IEEE Press (1994).