COMPUTER TOOLS TO SIMULATE ACOUSTIC PHENOMENA

Emilio Aramendia, Ricardo San Martin, Miguel Arana.

ABSTRACT
Acoustics is a branch of science whose mathematical models require a great effort of abstraction and spatial visualization. There are several pieces of computational software that allow us to represent these phenomena by means of graphical animations. Some of them enhance their capacity of calculation obtaining more rigorous graphical representations of the acoustic phenomena. Others enhance graphical functions making quicker animations. Software used for specific acoustic situations are also analysed. Very different software has been chosen for the comparative study, with a brief analysis of their operations and their potential as resources for teaching. Some of these have a very extensive scope of applicability, as can be *Mathematica* software. Some examples of animations that justify the scope of application of each program are illustrated.

KEYWORDS
Web-based learning, e-learning, acoustics, physical simulations, virtual learning environment.

INTRODUCTION

Some software systems relating to the teaching of acoustics are illustrated in this paper. The software analysed is divided into 4 groups. In the first group, software whose animations are created from the mathematical model that defines the acoustic phenomenon are described. Computer software such as *Mathematica* or *Matlab* belong to this group. In the second one, software whose animations are created knowing the graphical result of acoustic phenomena is detailed. Programs like *Flash MX* or *ImageReady* can be located inside this category. In the third group there is software that can simulate any acoustic phenomena from a Finite-Element Model, although this type of modelling can be more difficult and tricky, as shown by *Ansys 8* or *Fluent-Sysnoise*. Finally, in the last group, the software used for the resolution of specific acoustic problems is illustrated. These programs include *Raynoyse* and *Odeon* such as room acoustics simulation programs and environmental acoustic calculation software such as *SoundPlan* or *Mithra*.

MATHEMATICAL MODEL ANIMATIONS

In this section the animation tools of *Mathematica* are analysed in detail, showing some differences with other software that also belong to this group. The graphical tools of *Mathematica* are very powerful. Some of the animations found in the Web related to the didactics of acoustics and vibrations, are made in *Mathematica*. This program has a lot of functions for graphical representation in *2D* and *3D*; visualization of numeric data, vectors, surfaces, contours, graphical density and matrices, combination of graphs and *2D-3D* parametric representations. The combination with graphical functions allows us to obtain surprising results when representing *mathematica*l functions. From the didactics point of view, *Mathematica* is a useful software to simulate acoustic phenomena, combining the possibilities of graphical representations with animation functions. Fig. 1 shows an example of the capabilities of this kind of program. It represents the transmission and reflection of waves in cords. Only 14 code lines are needed to define both variables and functions. They can be programmed in an easy way. Just one more line is needed to create the animation. The function “Animate” makes a loop representing three graphical animations of functions versus time simultaneously. These functions represent the incident, reflected and transmitted wave respectively.

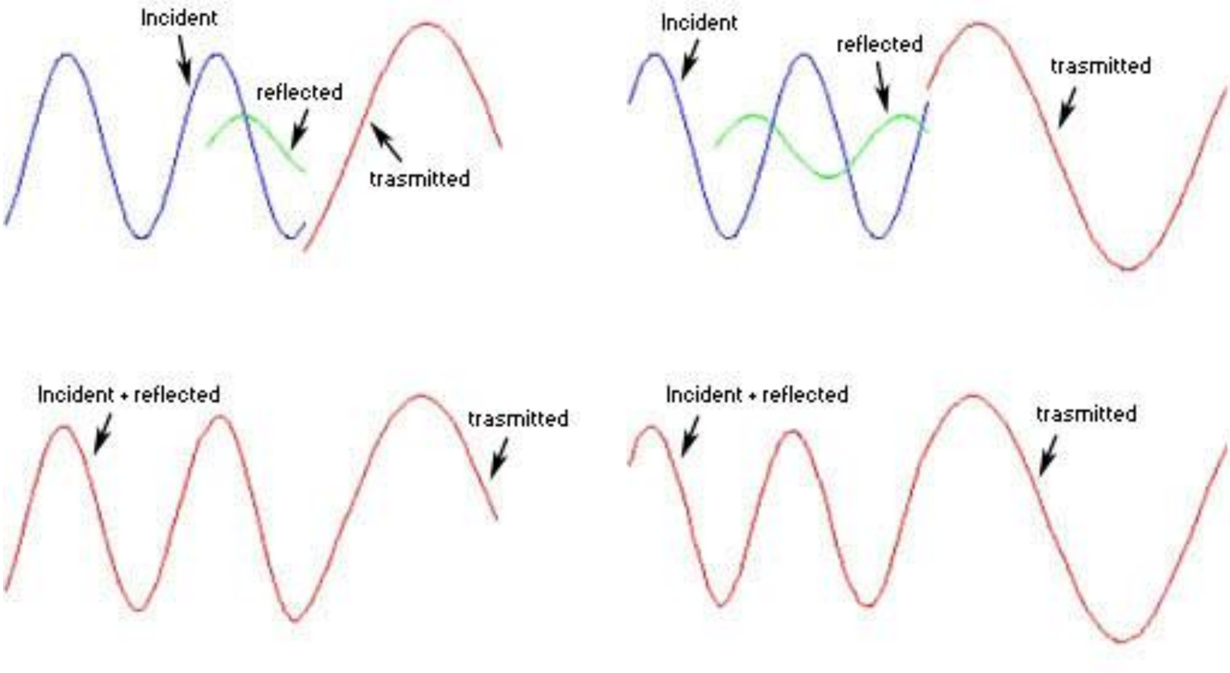

Figure 1. Intermediate frames of “trasmitted and reflected waves in cords” animation with Mathemathica. Reflected and transmitted waves at two different times (above) and resulting wave in which incident and reflected waves are added (below).

The function itself and its range definition can depend on the time variable. The animation can be controlled by changing the function throughout time and the time range in which this function is defined. According to this, the *Sin(wt-Kx)* function is an infinite sine moving towards the right at speed given by c=*w/k*. In order to see the edge of this wave it is only needed to control the time range in which the function is defined. This can be done by *Sin(wt-kx)* when *x<c*t*.

Other examples that show the advantages of *Mathematica* in graphical representation of physical phenomena are those related to the representation of eigenmodes. Combined eigenmodes (1,1) + (2,2) + (4,4) for a rectangular membrane is shown (Fig 2). The function that defines eigenmodes for a flat membrane is *Y= A·Sin(a·x)·Sin(b·z)·Sin(w·t)* where *A* is the vibration amplitude, *a* and *b* define the particular eigenmode and ‘*w·t*’ describes the harmonic character of the function.

The last program code line in the example of combined eigenmodes shown in Fig. 2 is: *Animate[Plot3D[y[x,z,t,w]+y1[x,z,t,w1]+y2[x,z,t,w2],{x,0,20},{z,0,20},PlotPoints->50,PlotRange->{{0,20},{0,20},{-12,12}}],{t,0,1,0.04},Axes->False]*.

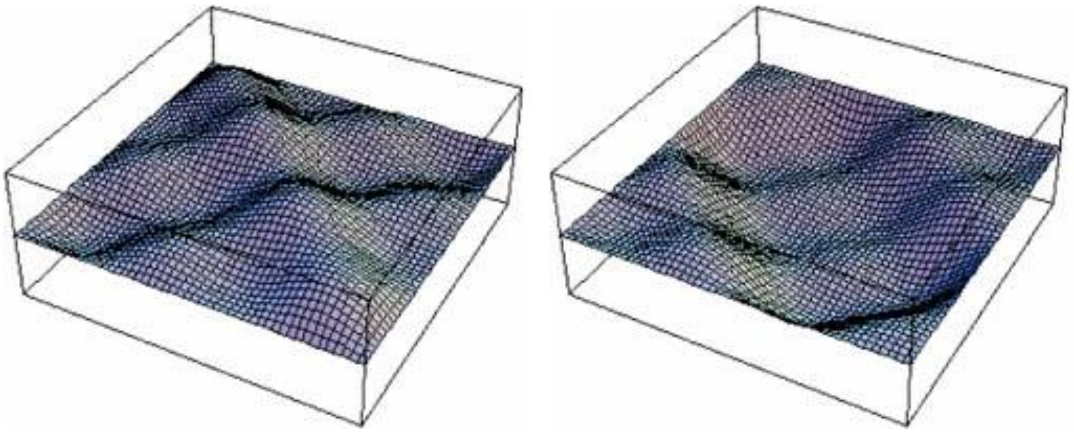

Figure 2. Intermediate frames of vibrations in membranes with Mathemathica. Combined eigenmodes *(1,1) + (2,2) + (4,4)*.

There are also other animation examples - circular membrane mode shapes, bar and cord mode shape - that can be programmed in a very similar way. The *Mathematica* workbook can be exported to an HTML format in an easy way. This is one of the *Mathematica* advantages from an e-learning and Web-based learning point of view. Nevertheless, although *Mathematica* gets the image of each animation frame, it lacks the last step, the creation of the animated GIF. The last step can be done using software described in the next section.

Matlab is another software package that can be included in this group. Working in *Matlab* is very similar to *Mathematica*. Each one has its own features, advantages and limitations. Graphical editor is more powerful in *Matlab* than in *Mathematica* but it is not integrated with animation functions. Also the definition of functions and matrices as well as working with them is more comfortable in *Matlab*. Nevertheless, graphical functions are clearer in *Mathematica*. In order to generate an '.exe' file that represents graphs from certain parameters, like interactive software of learning, it is needed to resort to *Matlab*.

GRAPHICAL MODEL ANIMATIONS

Sometimes it could be more interesting to show the physical phenomenon in a qualitative form, giving more importance to the concept than to the exact figures of the physical magnitudes that are being represented. In this section *Flash* software is analysed in detail. There are also other programs that can be included in this group such as *ImageReady*. *Flash* software can turn out to be useful when animations do not require excessive precision. In these cases, it can be done quicker with programs like *Flash*.

Sometimes *Flash* can also be used in combination with other specific software that show images but it is unable of creating animations from these. There are some programs that plot each image of the animation but do not join them in order to make the animation. Any frame needed can be exported to *Flash* making the animation with this software afterwards. Even if the software is not able to export images it is possible to screen capture. Software like *GIF Animator* or *Gif Movie Gear* could also be used for these actions.

Drawing tools in *Flash* are easy to use. They are based on the following concept: If an object overlaps another with a different colour, and it is moved later, the overlapped part of the first object is deleted. This characteristic determines the drawing strategy in *Flash*. *Flash* works with layers. It is possible to work independently in each layer, being able to show all or some of the layers. In each one of the layer there exists the time line. This time line contains all layer frames. *Flash* animations are created from graphical interpolation. There are two types of graphical interpolation, that of movement and the shape and contrast one.

The movement interpolation works in the following way: if we have an object in a position at the initial moment, and such an object is moved to other position in the final frame, we can create an animation of linear movement, by means of interpolation. For each intermediate frame the object shall be located at intermediate points on the line that links the initial and final points. There is a parameter that allows making non-linear interpolation, with time variation of speed, acceleration or deceleration. Another parameter exists that turns the object around its centre the times we want during the movement. There are also multiple synchronization options with sound and movement repetition. Any trajectory for the movement of the object can be described in an easy way.

Next are shown two examples of movement interpolation, used for the didactics of acoustic phenomena. The first example tries to show how the sound is focused on satellite dishes (Fig.3). Two different time sequences are shown. First, the reflection of rays travelling parallel towards the receiving antenna before rays affect the satellite dish and second the convergence of rays in the receiving antenna. It is only necessary to define the starting and final point of rays for each one of these sequences; the software interpolates the intermediate points afterwards.

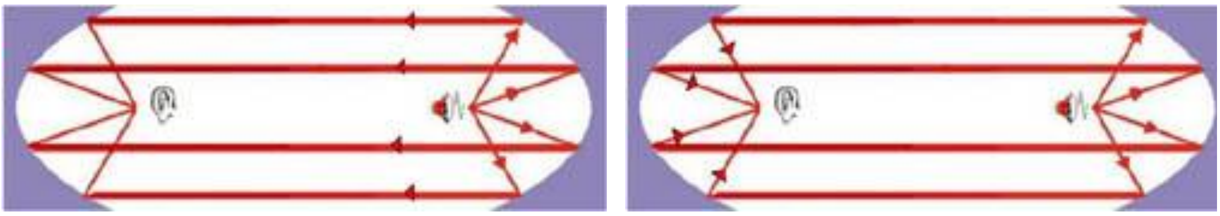

Figure 3. Intermediate frames animation of receiving antenna with *Flash*.

Another example of movement interpolation describing the trajectory is shown in Fig. 4. It represents the experience of the time interval discrimination between sounds in a tube. In this animation four time sequences exist, first the starting point before hitting the tube; second from the beginning until the ball arrives to the right ear; the third and fourth sequence corresponds to the remaining path of the ball that it has to arrive to the other ear. The speed of the movement is defined by the quotient between used distance to cross and frames used in the movement.

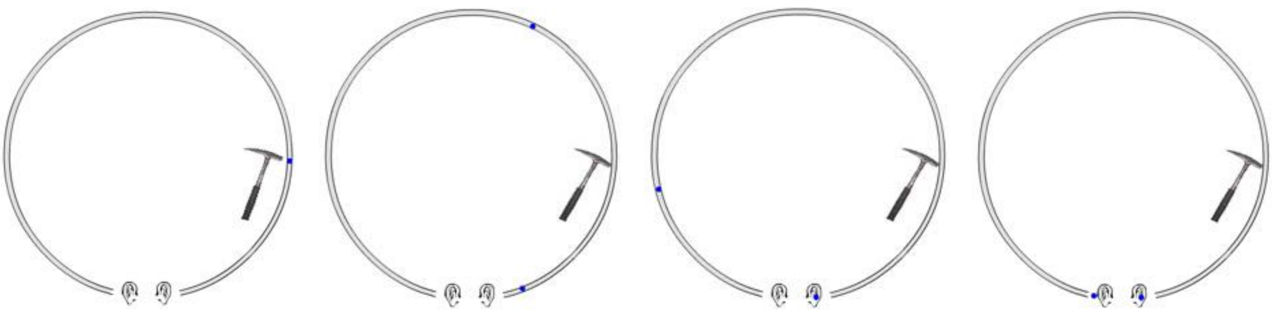

Figure 4. Intermediate frames animation of time discrimination tube in *Flash*

The shape and contrast interpolation is another tool to make animations with *Flash*. A progressive shape transformation from the initial to the final object can be done by simply defining the initial and final shape, since the software interpolates at intermediate frames. The interpolation can be done with a constant speed transformation or with certain acceleration-deceleration.

Fig. 5 is an example of interpolation by contrast. The animation represents the Helmholtz resonator's. Frame number controls the resonator's natural frequency. In the figure is represented pressure time-variation for 3 types of cavity at two random times. The length of the inlet neck varies for each type.

Figure 5. Intermediate frames of Helmholtz resonator animation with *Flash*.

Animations can be created at two levels: On the one hand, symbols that are individual animations and on the other hand, the scenes where more complex animations are created with these symbols. For example, a symbol representing a fly moving their wings can be created. Using this symbol in the scene is easy to create an animation of a fly flying. One of the most important advantages is the integration of *Flash* in HTML language.

FINITE-ELEMENT MODEL (FEM) SIMULATIONS

Finite element analysis software samples the physical environment to study with a mesh in which loads and border conditions are applied (forces, pressures, temperatures, speeds, deformations, electrical charges, etc.). The matrix of the system is assembled from these data allowing to arrange the differential equation matrix that defines the system. Later, the system is solved by means of the conjugated gradient of Jacobi method, frontal method, Spar method, etc, depending on the matrix type that defines the differential equation of the system. In mechanical systems, for example, once solved the system we can obtain both stresses and deformations in any mesh element in which the system was divided.

Most software of finite element analysis have separated the work stages in pre-processing, solution and post-processing. In this section *ANSYS* stages are shown in detail.

Pre-processing stage is divided in:
- Construction of the geometrical model.
- Definition of the physical properties to any different element type and association of properties to the geometrical model (type of material, behaviour, temperature, etc).
- Model mesh. In the finite element analysis software, model mesh is assembled with different unitary elements, which are selected depending on the system geometry, the physical properties and the type of analysis. The unitary elements are in the database. In the pre-processing stage the loads and the border conditions that define the problem are also included. Once we have set out the problem it goes to the following stage.

Solution stage:
- Different analysis types exist which depends on time load variance characteristic.
- Solution analysis is outside the pre-processing stage because the method used to solve the system is very different depending on the type of analysis. The *ANSYS* analysis types are: static, stationary, modal, harmonic, transient, spectrum, eigen buckling and substructuring. Each type of analysis has additional specifications; for example, in the harmonic case used in acoustic analysis, it is necessary to specify the frequency range to study and the subdivisions of this range.

Post-processing stage:
- The post-processing stage is the most interesting stage from the didactics point of view of the physical phenomena. The post-processing stage is out of the solution stage because many different graphical results can be shown from this equation once the solution of the differential equation is obtained. The processing time used to solve the system could be quite long; nevertheless the representation of the solution does not require much processing work. In this stage we must select the variable that represents our solution to the problem. In mechanical problems we could choose stress or strain results. Once the variable that represents our solution has been chosen, other

parameters have to be defined in the post-processing stage, such as the graphical type of variable representation. It can be shown displacements of the mesh, colour code, vectors, etc.

- *Ansys* is a finite elements simulation software oriented to the mechanical engineering that has been expanded in successive versions, introducing itself in other science branches such as electromagnetism, heat transmission, fluids and acoustics.
- From the acoustics point of view, *Ansys* 8 is sometimes unsuitable. Acoustics requires harmonic analysis. One solution for each frequency is needed in this process. This is the reason why FEM is not an efficient method to solve acoustic problems. Acoustics has a large range of frequencies to study. In new software versions the solid-fluid interaction problems have been improved.

Fluent is another software that can be included in this group. *Fluent* is a finite element analysis software focused on fluids. This software has better results in solid-fluid interaction and recently it has been united with *Sysnoise* for modelling and analysis of acoustic systems.

ROOM ACOUSTICS SIMULATION

In this section *Odeon* software is analysed, showing at the end of the section some differences with *Raynoise* software that also belongs to this group. *Odeon* is an acoustic room simulation software mainly based on ray tracing algorithms. The software is used fundamentally for concert halls designs. The software can be useful from the didactics point of view of acoustics, since it has several functions with interesting graphical representations. The steps to run a simulation are the following:

- First it is necessary to import the geometry from some CAD software, in .dxf format, for example. In this file the closed space we are going to study will be defined from the inner surfaces of the enclosure. All the surfaces - no matter how complex they are - must be modelled in *3D* with triangles and quadrilaterals by program requirement. The program has a geometry debugger that checks if surfaces overlap exists or if they do not join together.
- In the following step the sound sources and the receivers are defined and located into the room.
- Materials are assigned to the surfaces of the enclosure. Absorption coefficients to each one of the surfaces in frequency bands between *63 Hz* and *8000 Hz* are associated. Then the calculation parameters defining the type of ray tracing are set up.
- A receiver's grid that allows us to make a *3D* visualization of the acoustic parameters can be defined, with the surfaces coloured according to a scale representing the different values of the acoustic parameters at each point.
- Once everything is set up it is time to go to the work list and run the simulation. The software calculates the acoustic parameters of the room, energy decay curves, reflectograms and many others that can be shown. An auralization can be done, which is the solution to the binaural room impulse response.

From the didactics point of view of acoustic phenomena the software has some interesting functions:

- 3D ray tracing visualization: This software application allows us to visualize the ray tracing in 3D and see the successive reflections in the walls until a pre-established decay level. The number of reflections for each ray can be also selected. Distance and time of each ray travel can also be seen. We can either visualize all rays together or one by one. This tool can help us to understand how the sound spatial distribution is as well as how the enclosure type where the sound propagates depends on this sound spatial distribution. We can identify the most interesting surfaces either to reinforce or to reduce the reflected sound.
- Reflectograms: This tool shows the Sound Pressure Level of every ray versus arriving time. It can also be shown every ray azimuth and elevation. This tool can help us to understand and visualize echoes.
- 3D acoustic parameters visualization with colour scale: The values of acoustic parameters depend on the position of the receiver in the room. The visualization of the room in 3D using the colour scale to represent the acoustic parameters in each point is suitable for the acoustic evaluation of the room. Nevertheless, in order to understand the influence of geometry and materials in these parameters distribution a great knowledge on acoustics is required.

- Auralization: Perhaps auralization is one of the most interesting tools from the didactics point of view of the acoustics, because it allows us to hear how the room we are designing will sound. The technique is based on the Binaural Impulse Response (BIR) calculation. The convolution between emitted signal in the room (anechoic recording) and the impulse response, at any room position, will give us the sound that would be listened in that position. This tool can be useful to understand acoustical parameters for everybody even without technical education.

As *Odeon*, *Raynoise* is a program based on geometrical acoustic. Calculation of acoustic parameters can be done by tracing rays and image sources. The work stages are similar in both programs although some differences related to calculus algorithms exist. The most interesting thing from the didactic point of view is that *Raynoise* can show more acoustical parameters such as speech parameters.

ENVIRONMENTAL ACOUSTICS SIMULATIONS

Although there is some environmental acoustic simulation software, this paper only focuses on *SoundPLAN*. Some noise capabilities of *SoundPLAN*, related to the environmental noise evaluation, are analysed in this paper briefly. The geometrical model can be designed within *SoundPLAN* but generally another CAD software is used to make the *3D* geometrical model, exporting it to *SoundPLAN* afterwards. Both the noise sources and the receivers where SPL must be evaluated are modelled in the second stage. The definition of the source's characteristics such as spectrum power, directivity, etc., can be set up in a versatile way.

When the model is finished the calculating parameters must be set up. *SoundPLAN* is based on standards for road, railway, industry and aircraft noise. All major international standards have been implemented. *SoundPLAN* contains a source model calculating the sound power or a derived value from the traffic data for road, railway and aircraft noise. Industry noise requires the use of measured data. Different scenarios can be evaluated with different properties for the geometrical model, sources and receivers. Noise impact can be evaluated before the plant or railway is built. From a didactic point of view *SoundPLAN* is very interesting due to the graphic representation of sound propagation through the environment. With this tool it is easy to see how noise propagates through the environment. Flexible graphic tools for visualizing, mapping and presenting input and output data in a variety of mediums exist. The Grid Noise Map is generally used as the presentation tool. The influence of noise from the sources on the environment is evaluated for a grid of receiver points. Post-processing algorithms generate noise contour lines that are the basis of the evaluation of the environmental impact.

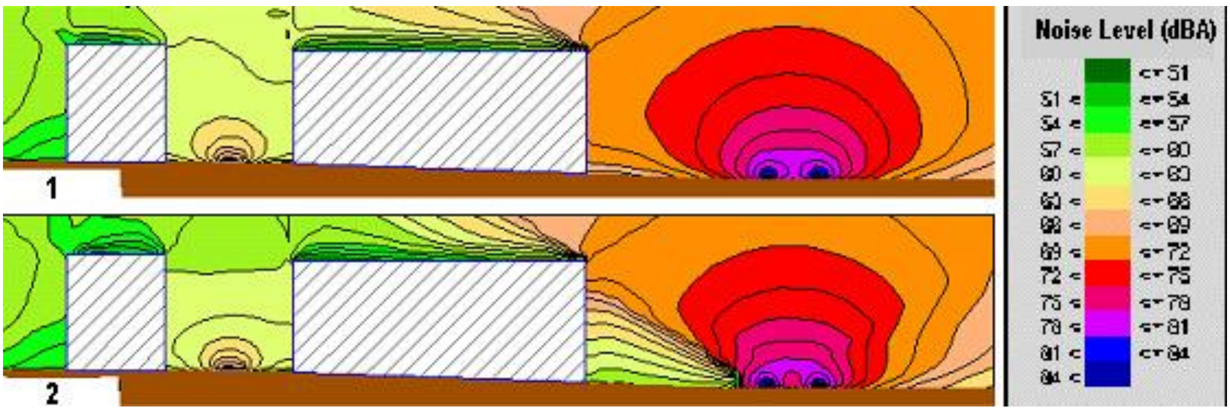

Figure 6. Noise isophones for transversal section of road from *SoundPLAN* with (2) and without (1) barrier.

SoundPLAN implements a very useful tool with regard to noise abatement. For residential areas near to a traffic road – when, in general, noise level limits are surpassed - the program is able to design a barrier to fulfil the noise criteria at facades. Furthermore, *SoundPLAN* designs the barrier piece by piece in order to minimize cost. An example of this tool is shown in Fig. 6 and Fig. 7.

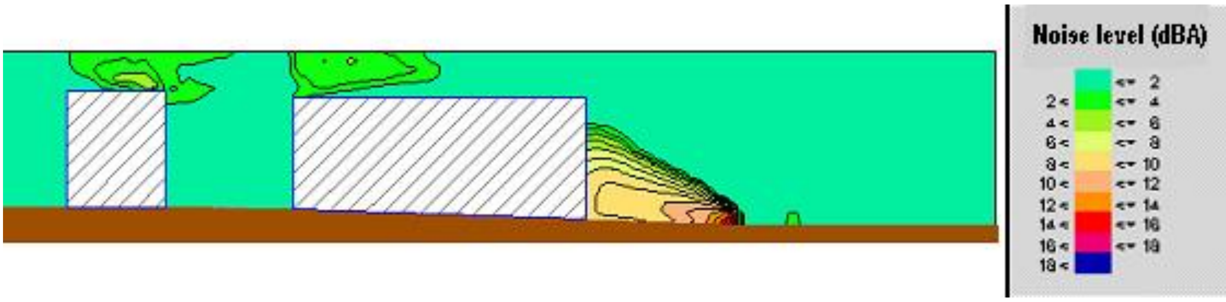

Figure 7. Noise abatement isophones for transversal section of road from *SoundPLAN*

SOFTWARE COMBINATIONS

The knowledge of graphical and calculation potentialities of different software allows us to choose the most suitable tool to make a suitable concrete task. There are many possibilities to combining them, simplifying the operations needed for a specific work. The object of this section is to show a practical example of program combinations, justifying why different programs have been used in spite of having a very concrete scope of application.

In the following example, sonic boom, the ray tracing requires certain precision (Fig 8). The components of vertical and horizontal speed of rays are different. According to this, the best program to make ray animation is *Mathematica*. Nevertheless, if we tried to make the complete animation with *Mathematica* we would have serious problems to include the airplane animation. We can finish the animation, including the movement of the airplane, exporting the ray animation from *Mathematica* to *Flash*, in a very simple form simplifying the steps.

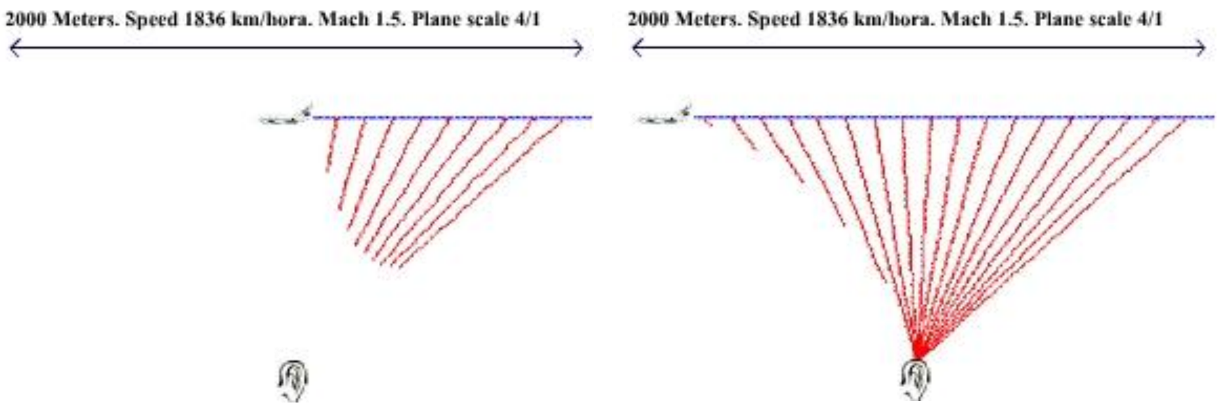

Figure 8. Intermediate frames of sonic boom animation.

CONCLUSIONS

The examples shown in this paper allow us to obtain an answer to the question of 'what software is more appropriate to make animations that simulate acoustic phenomena'. Some of the programs analysed in this paper have an extensive scope of applicability. We mainly focussed on animation tools. All examples of this paper can be done in many other software, but depending on the animation characteristics the selection of the software can minimize the working time. Knowing a large number of programs with animation tools allows us to improve the animation. Some of the animations have to be done combining different software in order to obtain good results. A rigorous animation often is not necessary to define the acoustic phenomena simulation. In those cases software such as *Flash* can be useful although this kind of software is not normally associated with physical simulations.

Some of the animations presented in this paper and many others can be shown in http://www.unavarra.es/organiza/acustica/index.htm (for the present, in Spanish). This web is inside Public University of Navarre site. The objective of this work is to introduce to our students in Web-based learning. Web animations help University students to a best understanding of acoustic concepts whose mathematical formulae are often more difficult.

REFERENCES

Kinsler, L.E.; Frey A.R.; Coppens, A.B.; Sanders , J.V.(1999). Fundamentals of Acoustics. John Wiley& Sons Inc., N.Y.

Mathematica 5. © Copyright 1988-2003 Wolfram Research.

Matlab 6.5.1 © Copyright 1988-2003 The MathWorks,Inc.

Macromedia Flash Mx. © Copyright 1997- 2002 Macromedia.

ImageReady 7.PhotoShop. © Copyright 1990- 2002 Adobe Systems Incorporated.

Ansys 8. © Copyright 2003 Ansys Incorporated.

Odeon 6.01 © Copyright 1988-2003 Odeon A/S Denmark.

Sound Plan 5. Copyright © 1986-2004, Braunstein + Berndt GmbH. All rights reserved. US Copyright © 2001-2004, SoundPLAN LLC.

http://www.gmi.edu/~drussell/Demos.html

http://www.unavarra.es/organiza/acustica/index.htm

E. Aramendia
Research Assistant
Physics Department
Public University of Navarre
Campus de Arrosadia, s/n
31008 Pamplona. Spain.
Tel.: 34 948 168451
Fax: 34 948 169565
E-mail: emilio.aramendia@unavarra.es