Java Programming – Gravitational Waves

Daniel Little, Valentin Miu, Ruairidh MacLeod

**Abstract**. This report presents the joint creation of a 3D environment capable of visually representing gravitational waves emitted by a binary pulsar system, as well as deducing their effect on a Laser Interferometer Space Antenna type detector. It briefly explains the physics and object-oriented programming concepts behind the program, shows the libraries and tools used, and presents the features of the software.

1. Introduction

We were commissioned to create a Java program illustrating some research project currently being carried out at the university. After investigating the work of the different research groups, we decided that the work being carried out by the Institute for Gravitational Research on the LISA project was the research that would benefit the most from a simulation that would simplify the concepts for those with less of an astrophysics background.

One of the great issues the scientific community tackles today is that of gravitational waves. These waves represent ripples in the fabric of spacetime, generated by accelerating masses. Being a prediction of Einstein's theory of general relativity, detection of gravitational waves would be crucial proof of the validity of this theory.

The LISA (Laser Interferometer Space Antenna) project aims to build a spacebourne gravitational wave observatory, which will target sources in the low frequency band and compliment land based detectors like LIGO (Laser Interferometer Ground Observatory) that already provide data from the high frequency band. LISA was originally intended to be a joint venture between NASA and ESA, but budget restraints caused NASA to pull out, resulting in a revised project with a smaller detector called eLISA (evolved LISA) set to launch around 2034. [1]

Our program simulates gravitational waves emitted from a binary star system and the eLISA detector reacting to them.

1. Physical background
   1. *Gravitational waves*

Einstein first postulated the existence of gravitational waves in 1916 as a consequence of his theory of general relativity [2]. Gravitational waves are propagating fluctuations of gravitational fields or “ripples” in spacetime, generated by very massive bodies moving very fast the waves travel outwards from their source at the speed of light. Every body in their paths will experience a tidal gravitational force, this force acting perpendicularly to the direction of propagation and changing the distance between spatial points.

Gravitational waves are interesting to scientists because of the potential information they carry about their sources, as opposed to their EM counterparts. Their adequate detection would open up a whole new window for observational astronomy complementing our current view of the universe and helping us to directly observe the spacetime around black holes and the merging of binary systems. Direct proof of gravitational waves would also help to verify Einstein’s theory of general relativity and further our understanding of the fundamental laws of physics.

Out of the four types of fundamental force, gravity is the weakest, and thus the hardest to measure or observe. As a result, the detection of gravitational waves requires large masses (to generate waves of sufficient amplitudes), and extremely sensitive detectors (to pick up infinitesimal variations in the fabric of spacetime).

A good candidate for the first requirement are binary systems of hyper-dense astronomical objects, such as pulsars or black holes. The generation of gravitational waves by such systems has already been shown indirectly by Joseph Taylor and Russell Webb; they achieved this by observing the decay of the orbits in a binary pulsar system [3]. Since the emission of gravitational waves evidently corresponds to a loss in the system's energy, this decay is explained by general relativity better than most of the competing theories.

|  |
| --- |
| What are Gravitational Waves? Earlier in the year, physicists made headlines claiming that they may have detected gravitational waves—only to announce, some time later, that it had just been dust. How could such a blunder be made? How could their detection equipment not immediately distinguish between a fundamental discovery and some specks of debris? Turns out: pretty easily. Gravity is the force that affects us every second of every day, even if we’ve forgotten to notice it, but it’s by far the weakest of the four fundamental forces (gravity, electromagnetism, the weak nuclear force, and the strong nuclear force). You can test this easily at home—pick up a piece of metal with a magnet, or some scraps of paper with a negatively charged comb. As well as being the weakest, gravity is perhaps the least understood. Two fundamental ideas have been proposed: Newton explained that gravity is an attraction between two bodies, then Einstein’s General Relativity modified this explanation, postulating that matter actually causes warps in space-time, like balls sitting on a rubber sheet, and this distortion is felt as gravitational influence. As an object with mass moves, the curvature of space-time changes accordingly to remain ‘around’ the object. If an object accelerates, it can cause ripples in the curvature of space-time—but only if its motion isn’t perfectly spherically symmetric. Consider a supernova: if the star was exactly spherical, when it explodes it will not produce a gravitational wave, but a star that is even slightly asymmetrical will. These disturbances propagate outwards as a wave. (Check out some potential sources of gravitational waves here.) We’re familiar with the idea that the electromagnetic force travels as a continuous wave, with an electric and magnetic component propagating at right angles to each other, transporting energy as electromagnetic radiation.  (Image Credit) In a similar way, we expect gravity to be expressed as a gravitational wave, though these waves aren&#8217;t oscillations of electric and magnetic fields: they&#8217;re oscillations of space-time. These waves would have a maximum speed of c (the speed of light), so technically, if our Sun suddenly disappeared, the Earth would keep happily orbiting for another 8 minutes, until the gravitational ‘information’ reached us. But here’s the issue: though gravitational waves are supported by mathematics, we haven’t actually been able to observe any yet. Their intensity drops off as it get further away from its source, so by the time they reach Earth, they are predicted to be very small, with frequencies in the range of 10-16 to 104 Hertz. For decades, researchers have built and worked on ever-more-sensitive detectors, but since gravitational waves are weak, there is a huge amount of interference to be weeded out, and no definite detections have been made yet. The European Space Agency is currently developing a space-based gravitational waved observatory called LISA (Laser Interferometer Space Antenna), which would eliminate a lot of interference, but unfortunately may not be launched for decades. Detecting gravitational waves would be revolutionary, not just because it would confirm theories about general relativity and the nature of gravity, but because gravitational waves can penetrate parts of space that electromagnetic waves can’t. In the future, we could use gravitational wave ‘telescopes’ instead of optical and radio telescopes, and have a new ‘view’ of exotic objects like black holes and of times near the very beginning of the universe. As for the scientists whose breakthrough turned out to be dust—well, what can you expect from a telescope named BICEP? |
| Figure 1. A graphic representation of a binary black hole system generating gravitational waves. Credit: CSIRO. |

* 1. *Detection of gravitational waves*

The first attempt at direct detection was in the mid-1960s by Joseph Weber. He designed and constructed heavy metal bars that were seismically isolated. To these bars he attached a set of piezoelectric strain transducers in such a way that the bars could detect the vibrations due to gravitational waves that could incidentally pass through them. Today there are a number of such devices operating at much higher sensitivities than Weber's, but these modern ones are still not quite sensitive enough to detect these waves.

Another way to detect small variations in spacetime is by laser interferometry. Assume a detector with two very long arms, equal in length, forming an angle. At the intersection of these two arms, a laser beam is split in two (each half evidently having the same phase), and each of the two resulting beams are sent down a different arm. These beams are reflected at the end of their respective arms, and sent back to their intersection point, where they interfere, forming a pattern. For greater accuracy, each beam can be reflected back, undergoing several trips before being collected and their interference pattern observed.

Assume a gravitational wave incident on the detector. Due to the differing orientations of the two detector arms, it will create a very slight fluctuating difference in the length of these arms. As a result, the interference pattern of the laser beams will change, ideally enough to be observed.

|  |
| --- |
|  |
| Figure 2. Diagram of the LIGO detector. Credit: Caltech. |

Unfortunately, gravitational waves of great enough amplitude require extremely large masses, undergoing great acceleration, at a reasonable distance from the detector. Since such sources are not immediately available, detectors require immense arm lengths, and protection from the slightest unwanted perturbations. Seismic effects make this impossible anywhere on the planet.

The obvious solution to this is a space-based detector. The LISA (Laser Interferometry Space Antenna) is planned to be such a detector. Constructed from three satellites forming an equilateral triangle, at great distance from each other (about 1 000 000 km), the ensemble will share the Earth's orbit around the Sun, with a certain degree of orbit phase retardation. The satellites work similarly to the aforementioned laser interferometry detector (one satellite is the laser emitter and detector, the other two are mirrors); they will be kept in formation with high-precision thrusters. A proof-of-concept small version launch is to be conducted in late 2015; the real version has a tentative launch date of around 2034. [1]

* 1. *Mathematical properties of a gravitational wave*

Gravitational waves are very complex, and to describe them mathematically is a long and difficult process. But they can be simplified significantly; at least enough that they can be simulated in our Java program.

Since all gravitational waves we can expect to pass through the earth are very weak, a linearized theory of gravitational waves should describe them accurately. What this means is that we can assume that the observer is far away, the area of spacetime in which he exists being described by a metric *gμv*. Any change in the matter distribution of this metric can be described by:

(1)

where *hμv* is a tensor describing the variations induced in the spacetime metric i.e. gravitational waves. [4]

To solve *hμv* we have to solve Einstein’s equations for a varying matter distribution. This is not particularly easy, but there is a way we can proceed that greatly helps us. This is to assume that *hμv* is small (|*hμv*|<< 1); because of this we need only assume that the terms in *hμv* are linear in our calculations. What this means is that we are assuming that the disturbances produced are not huge, so by using this *linearized approach* we can get accurate results for weak field assumptions.

The first attempt to prove that these gravitational distortions propagate as waves travelling at the speed of light was by Einstein himself. He proved that by assuming linearized perturbations around a flat metric (i.e. *gμv* = *nμv*), it results that the tensor is governed by a wave equation which admits plane wave solutions similar to the ones of electromagnetism. Here is the gravitational field:

(2)

This is the three-dimensional wave equation. By simplifying this using Hilbert's gauge condition (equivalent to the Lorentz gauge condition of electromagnetism) we can obtain the simplest solution to the wave equation shown below:

(3)

where is the polarization tensor and is the wave vector; in physical applications we will only use the real part of the above wave solution.

Up to this point has had six arbitrary components, but due to the gauge of freedom of one of the vectors in this can be reduced to only two in the suitably chosen gauge. An example of such a gauge is the transverse-traceless or TT-gauge, in this gauge only the spatial components of are nonzero.

Using this TT gauge Einstein derived the quadrupole formula for gravitational radiation. The formula states that the wave amplitude *hij* is proportional to the second time derivative of the quadrupole moment of the source:

(4)

Where is the quadrupole moment (shown below) in the TT gauge evaluated at retarded time *t-r/c* and *ρ* is the matter density in a volume element *d3x* at the position *xi.*

(5)

1. Program development

Creating the software presented three main problems: the physics equations, the structure of the program, and the actual code. Although each of these was the main concern of a different group member, some more complicated issues were resolved by the group as a whole.

* 1. *Object-Oriented Programming*

In software design, an object-oriented language is a paradigm where everything in the program is modelled as a virtual object. This is distinct from a procedural language, one in which code is executed line by line and is separated into different functions. An object in this sense is a virtual construct that represents some part of the program as a whole. In our program examples of objects would be the detector model, the window we are creating the 3D content in, or an object representing the camera in 3D space. In general, an object is an encapsulated piece of code that has some *state* and provides some *services* to other objects that interact with it. Objects communicate and alter each other’s state using these services. Using the example of an object that represents a vector, its state would be its length or direction, and services that the vector object provides would allow another object to change the length or direction of the vector. The concept of object-oriented programming is a very powerful method for designing and visualising software that would normally be very complex and intangible.

Java has been designed as an object-oriented programming language, and as such we utilised concepts of object-oriented programming in the development of our application.

* 1. *Model View Controller*

In object-oriented programming, a design pattern is a well proven and reliable solution to common problems in software design. A design pattern can break down an initially complicated problem into well-defined parts. We have used the Model View Controller (MVC) design pattern to help structure our code, since it is well suited for use in graphically rich application such as ours. In MVC, the main program class files are split into three distinct objects:

* Model – Contains the data describing the current state of the program, updates the View when its state is altered.
* View – Generates all the graphical output that the user sees.
* Controller – Handles user input and updates the controller.

|  |
| --- |
| http://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/MVC-Process.svg/500px-MVC-Process.svg.png |
| Figure 3. Interaction between different components of the MVC pattern. Source: Wikipedia. |

* 1. *Tools and code writing*

The program was written using the Eclipse development environment, … that supports the use of some of the tools required in the production of the software. In order to create the graphical user interface, a specialized library was required; Java currently offers JavaFX and Swing for such purposes. Due to its 3D graphical capabilities, available since version 8 (2014), JavaFX was chosen. Along with the GUI containers and controls required for user input, JavaFX also allows the creation of 3D scenes with simple or custom meshes, UV texturing, simple Phong shading and object animation.

|  |  |
| --- | --- |
|  | |
| **Figure 4.** Example JavaFX 3D scene. |

The program is formed from a Main class (the class which is actually run), the Controller class (which initializes the GUI and 3D scene, and contains the functions that the interface components are linked to), as well as several classes defining custom objects in the scene. Finally, an FXML file (an XML-style file for JavaFX GUI creation), was used to define and position the interface components, as well as link them to their specific functions in the Controller class.

In order to facilitate the creation and layout of the interface, JavaFX Scene Builder 2.0 was used. This allowed the visual positioning of the containers and controls, as well as the input of their relevant parameters. The FXML file was generated automatically by this program.

|  |  |
| --- | --- |
|  | |
| **Figure 5.** The JavaFX Scene Builder 2.0 environment. |

In addition to the choice of 3D graphics package, a version control system (VCS) was also utilised. A VCS allows a team of developers to concurrently work on a piece of software and controls the entire history of the software across every iteration. Using such a system helps to avoid some common issues in software development such as two people trying to update a file at once or a file accidentally being deleted. We chose Subversion as our VCS. This was motivated by the fact that it is available as a plug-in for Eclipse, and it is simple to operate.

1. Program features
   1. *Binary system parameter input*

The program assumes a binary system with circular orbits. Although orbits of celestial bodies are generally elliptical, binary pulsars tend to circular orbits when relatively close to coalescing (when the gravitational waves emitted are strongest). Thus, circular orbits are the most relevant to this simulation, but elliptical orbit support may be added later on.

The relevant parameters, such as the masses of the pulsars and the distance between them, are input by the user in their respective text fields in the *Stars* tab. The radii of the pulsars are also settable, and the user may choose to force their densities to be equal (that is, the cubes of the radii are proportional to the masses). The program uses these parameters to determine the orbit distances and periods; since the periods may be exceedingly large or small, a time stretch function (for slowing down or speeding up perceived time) is included.

The units of all distances and angles, for all relevant components of the scene, are settable to avoid exceedingly large or small numbers as much as possible.

Since the distance between the system and the detector can be expected to be many orders of magnitude larger than the size of the system itself, the scale of the system can be overridden to ensure it is properly visible regardless of the distance of the viewpoint (that is, the 3D scene camera). The scaling factor can be determined automatically, and can also be linked to the viewpoint distance so that the system's perceived size is roughly constant. If the system hinders the observation of the gravitational wave pattern, it can be hidden altogether.

|  |
| --- |
|  |
| **Figure 6.** The binary system and the *Stars* tab. |

* 1. *Gravitational field visualization*

In the *Field* tab, the user can choose between two methods of visualizing the waves. The first represents the amplitude of the waves over space and time as a colour map; black and white represent points of maximum space expansion/contraction (relative to the distance from the system), and greys represent points of lesser such distortion. The x and y size and resolution of the colour map, as well as its angular orientation, can be set by the user.

The second method shows an array of vectors (represented as 3D arrows of variable length and orientation), which indicate the magnitude and direction of the gravitational field at the points of the array. The x, y and z dimensions of the vector array, as well as the x, y and z distances between the array points, can be set by the user, along with the angular orientation of the array.

|  |  |  |
| --- | --- | --- |
|  | colormap | |
| **Figure 7.** Gravitational waves visualized by the program as a vector field (left) and as a colour map (right). Note that the implementation of the vector field is not yet complete, and is not physically accurate here. | |

* 1. *The detector*

The LISA is represented by three grey boxes, two of which are connected to the third by thin red cylinders (indicating the detector arms). The angles and lengths of the arms are settable, along with the global position and rotation of the detector. The number of trips the laser beams take through each arm can also be set by the user. The program calculates the difference in arm length at each moment, and determines the interference pattern. Similarly to the system, the detector has a scale override, for the same reason.

The approximations used in the equations do not allow for an exact result if the detector is far away from the plane of the system. However, since the gravitational distortions rapidly become very small as the distance from the plane increases, the zone around this plane is the main area of concern.

|  |
| --- |
|  |
| **Figure 8.** The detector and the Detector tab. |

* 1. *The camera*

The observer viewpoint (the position and rotation of the camera) is controlled by the mouse or the keyboard, or by entering the relevant coordinates numerically, using the respective text fields in the *Camera* tab. A list of mouse and keyboard commands can be brought up from this tab.

Since the viewpoint distance can vary by many orders of magnitude, the units used by the mouse and keyboard controls are settable.

1. Conclusions

Using the approximate general relativity equations mentioned, the resulting program should be good enough to be a useful educational tool, showing both the form of the gravitational waves produced by a binary pulsar system, and their effect on a LISA-type detector.

References

1. http://www.esa.int/Our\_Activities/Space\_Science/ESA\_s\_new\_vision\_to\_study\_the\_invisible\_Universe
2. Einstein, A (1918). "Über Gravitationswellen". Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften Berlin. part 1: 154–167
3. Weisberg, J M and Taylor, J H. "Relativistic Binary Pulsar B1913+16: Thirty Years of Observations and Analysis"
4. Kokkotas, K D. "Gravitational Wave Physics"