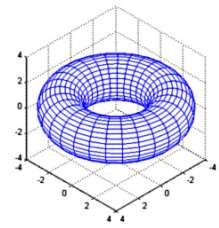


Simulating gravitational field near a torus

Description	Fitting a 2D model on data sampled from 3D space
Period	Spring 2016
Languages & Libs	Matlab
Tags	Applied mathematics



There are many games with a strong emphasis on gravity, and at times even multi-body trajectory simulations. However they all seem to be based on spherical geometry (as planets are shaped by gravity), but other shapes should create interesting trajectories. As torus has rotational symmetry its gravity field can be modelled on a 2D cross-section. In this project torus' field is estimated in 3D, projected to 2D and interpolation functions are fitted. The space- and time-efficient model could be used in a game to do physics simulation in real time.

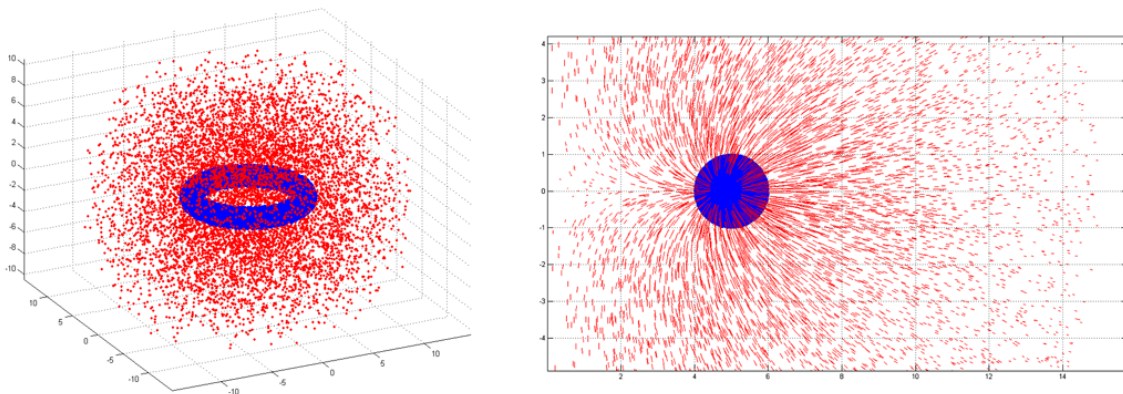


Figure 1: Sampling random points within the torus and the space outside it (on left). Due to rotational symmetry they can be projected and repressed on a 2D cross-section (on right).

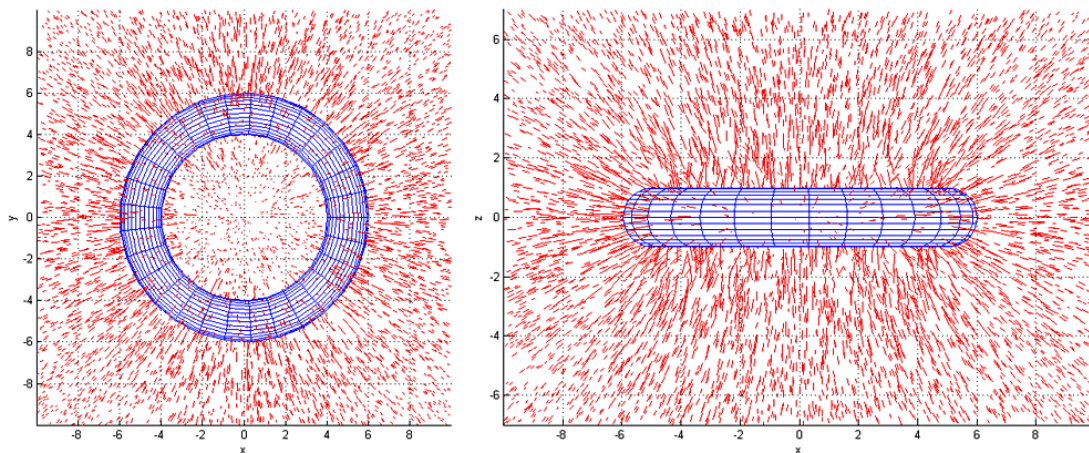


Figure 2: Visualization of gravitational force field around the torus, shown in XY and XZ planes.

Instead of using numerical integration this simpler method was used as I wasn't sure I wouldn't introduce unintended biases when defining the integral over the volume of a torus. The process starts from generating random points within the torus (shown as blue in Figure 1) and outside it (shown in red). For each point outside the torus the gravitational field's force vector is estimated. These are then projected to a 2D cross-section, as shown on the right side of Figure 1. 3D plots of fields are shown in Figure 2. It can

be seen to decay towards zero (as expected) at greater distances, but interestingly field goes to zero also at the origin due to the symmetry.

After 3D coordinates have been rotated to the 2D cross-section plane, coordinates are transformed log-polar coordinates. These are shown in Figure 3 (forces along x-axis) and Figure 4 (forces along y-axis). Their cartesian-coordinate counter-parts are shown in Figure 5. Shape of these graphs is a bit more apparent if they were seen in a rotating 3D animation. Red dots are correct values and blue values are forecasted by the fitted interpolation model. There is some residual error but it shouldn't be apparent in a game-like simulation.

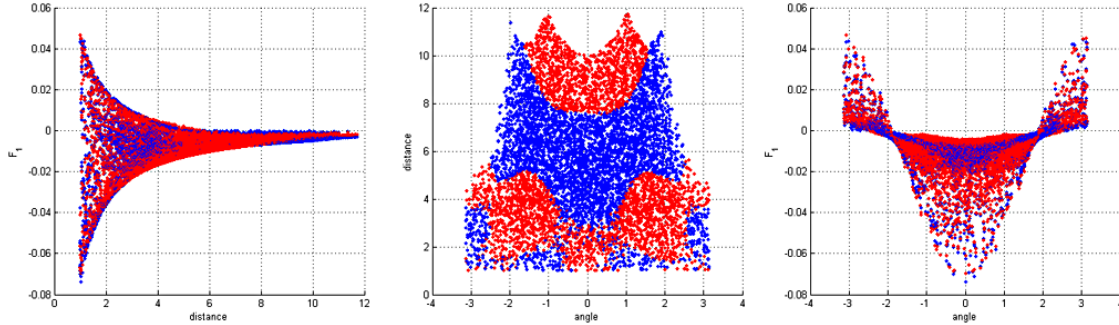


Figure 3: Gravitational field's force along the x-axis (in cross section coordinates).

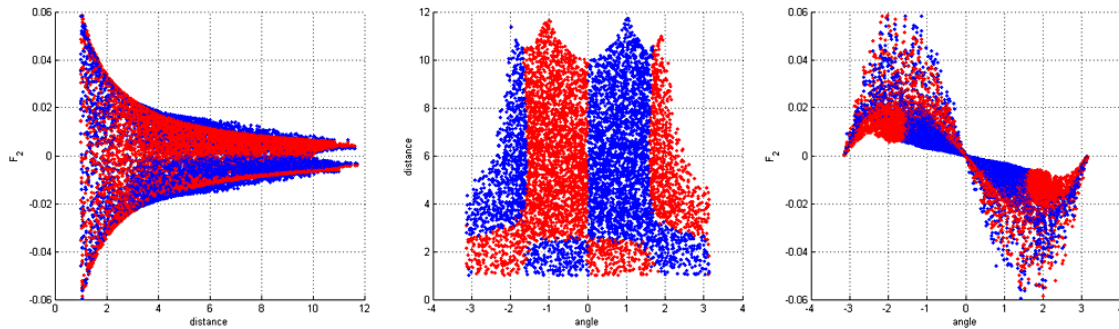


Figure 4: Gravitational field's force along the y-axis (in cross section coordinates).

The model for the force along axes have the following forms:

$$F_x(\alpha, d; p) \approx (p_1 \cos(\alpha) + p_2 \cos(2\alpha) + p_3) \cdot \exp(p_4 \cdot d^{p_6}) + p_5 d + p_7 \quad (1)$$

$$F_y(\alpha, d; p) \approx (p_1 \sin(\alpha) + p_2 \sin(3\alpha)) \cdot \exp(p_3 \cdot d^{p_4} + p_5 \cos(\alpha) + p_6 \cos(2\alpha)) \quad (2)$$

Values for p at these two functions have the following values:

- F_x : [-2.1924 0.1201 -0.3083 -3.6119 5.4967e-04 0.2802 -0.0073]
- F_y : [-15.0000 -1.4790 0.1457 -5.5883 -0.2798 -0.1553]

This geometry creates a field which have some interesting properties, when compared to the "standard" inverse square law ($1/r^2$) of a spherical planet or a star. Four examples of these are shown in figures 6 - 10. There seems to exist various classes of non-closed trajectories as one could say torus has two degrees of "freedom" instead of just the one on spheres. I was happy to discover that there exists orbits which go through the hole in the middle (figures 9 and 10).

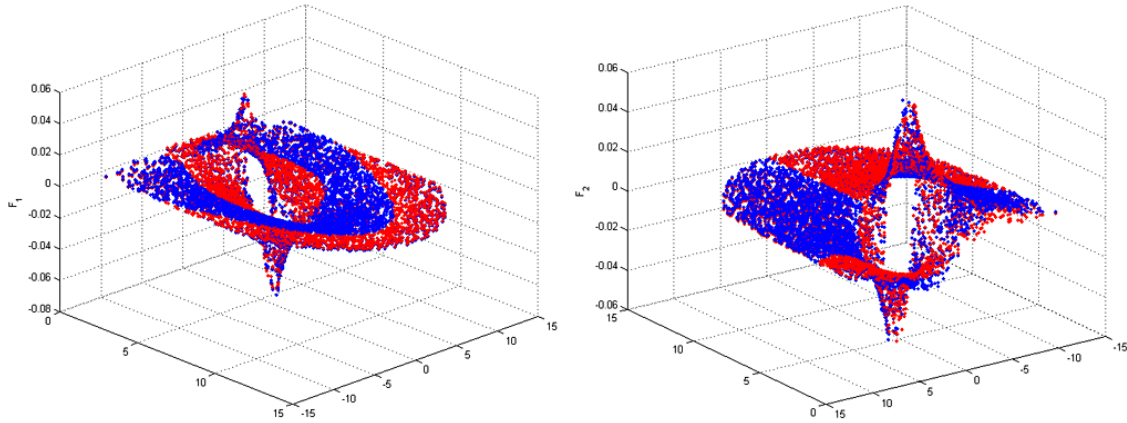


Figure 5: Force vectors' x- and y-axis components plotted in cartesian coordinates.

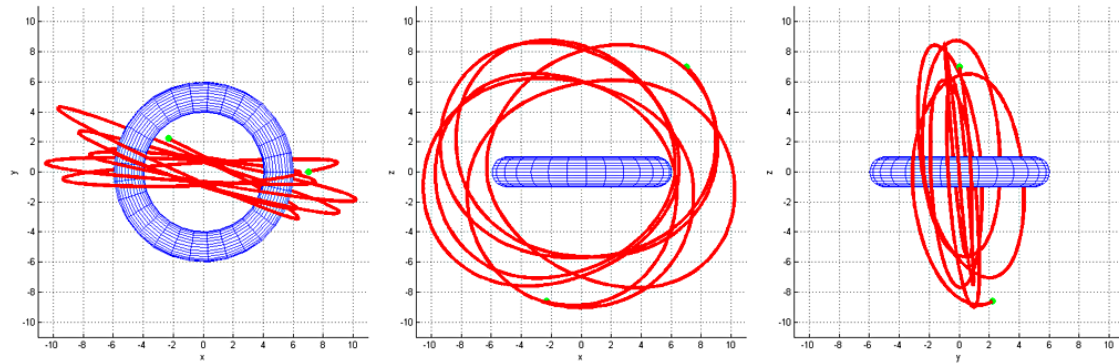


Figure 6: A quite circular trajectory on the XZ-plane.

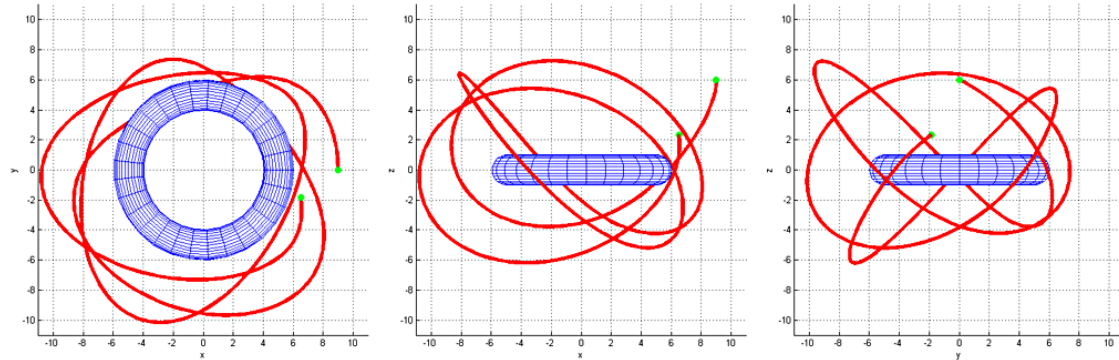


Figure 7: A moderately chaotic trajectory, going back-and-forth on all dimensions.

Trajectories were estimated by running the differential equation through the standard [midpoint method](#) with a suitably small time-step. Most trajectories seem to be fairly chaotic and unstable, so simulating additional bodies such as moons might have non-interesting results. It would be interesting to simulate an atmosphere consisting of ideal gas but I have no prior experience on that even on a spherical body. It would also be worth studying game-mechanics on the FPS genre, and why not also for a platformer?

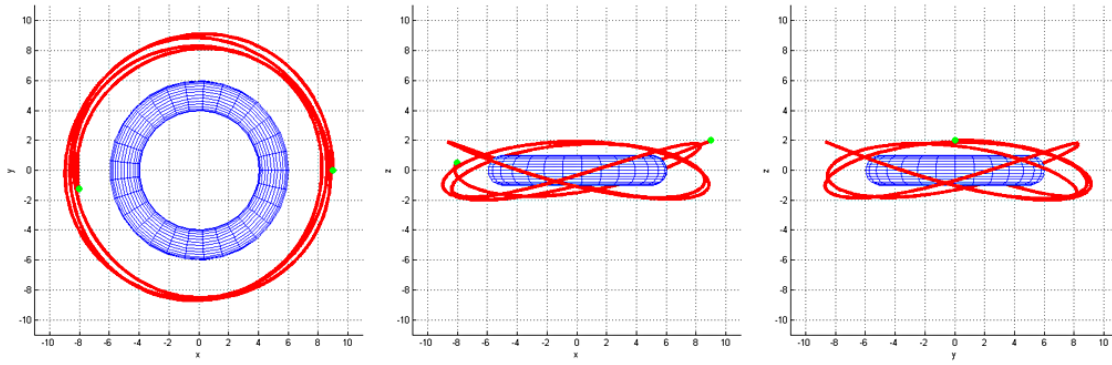


Figure 8: A circular trajectory on the XY-plane.

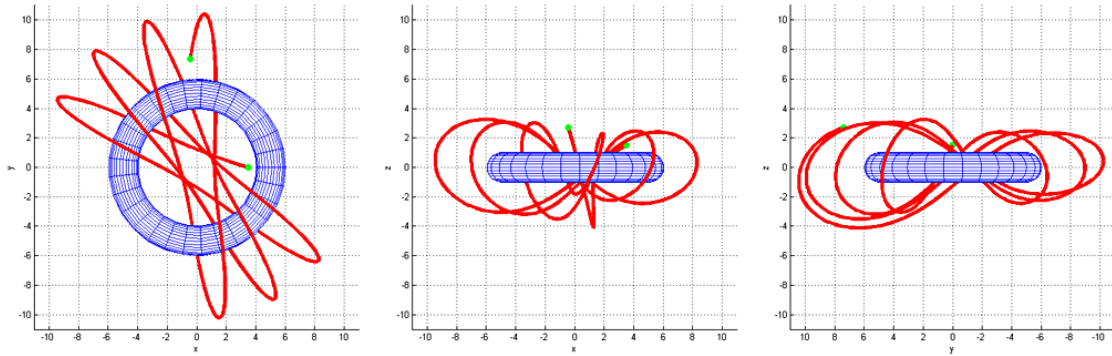


Figure 9: An interesting trajectory which goes through the torus' hole at the middle.

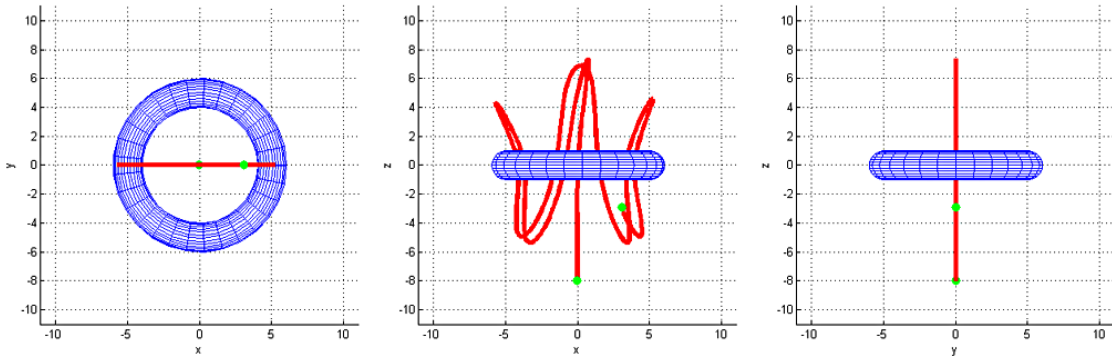


Figure 10: An oscillating path on the XZ-plane.