Project 4: Ants and triangles

Figure 1: The ant

1 Introduction

In this project we will consider a problem proposed on project Euler (`https://projecteuler.net`). It is problem 613.

1.1 Problem Statement: Project Euler 613

Dave is doing his homework on the balcony and, preparing a presentation about Pythagorean triangles, has just cut out a triangle with side lengths 30cm, 40cm and 50cm from some cardboard, when a gust of wind blows the triangle down into the garden. Another gust blows a small ant straight onto this triangle. The poor ant is completely disoriented and starts to crawl straight ahead in random direction in order to get back into the grass.

Assuming that all possible positions of the ant within the triangle and all possible directions of moving on are equiprobable, what is the probability that the ant leaves the triangle along its longest side?

2 Programming task: a Monte Carlo approach

The Monte Carlo approach is a method that computes probabilities by repeated simulation of a random event. It is very robust and can almost always be applied to compute statistics for a given phenomenon, however to get a good accuracy requires a large number of random samples as we shall see below. We will consider a slightly more general problem where we explore the above mentioned probability for a set of Pythagorean triangles. The program should execute the three following tasks

1. Read the triples (a, b, c) corresponding to a (primitive) Pythagorean triple from the file `triangle_triples.data`

2. for each triple compute the probability that the ant exits across the longest side. The Monte Carlo method suggests the following brute force approach:

   (a) fix a sample size N

   (b) using the function `random()` (in the module "random"), generate a position (x, y) and a direction given by an angle θ.

   (c) find the exit side of the ant starting in (x, y) and moving in the direction given by θ.

   (d) increase the counter for this side.

   (e) start again from (b) until N experiments have been done

   (f) for each side return the number of ants that have exited through it, divided by N.

3. find the triangles for which the probability of exiting through the longest side is the smallest and the greatest. Give the values and the triples of the associated triangles. Check if your result changes as you increase the sample size N. Report a histogram showing how the Pythagorean triangles in `triangle_triples.data` are distributed over the different exit probabilities. (For details on how to plot a histogram see `https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.histogram.html`. There is also an example code `histogram_demo.py` on the moodle page.

3 General comments

- The convergence of the above algorithm to compute the probabilities is rather slow. Make sure that N is sufficiently large so that you can trust your approximation to two decimals.

- If you want to generate Pythagorean triples yourself you can use *Euclid's formula*: for m, n integers $m > n > 0$ let

$$a = m^2 - n^2,\ b = 2mn,\ c = m^2 + n^2.$$

  Then (a, b, c) for a Pythagorean triple.

- On the project Euler page they ask for a result rounded to 10 digits after the decimal point. Can you achieve this precision with your method? A geometric argument can be used to find the probability of exiting along the longest side for each point in the triangle. The total probability is then obtained by integrating this expression over the triangle and normalizing with the triangle area. The exact probability of exiting through the longest side of the $(3, 4, 5)$ triangle is

$$\frac{12\pi + 16\log(4) + 9\log(3) - 25\log(5)}{24\pi} \approx 0.3916721504. \tag{1}$$

  You can validate your code by checking that the approximate probability for the triangle $(3, 4, 5)$ indeed appears to converge to this value as N is increased.

4 What to submit for the assessment

You should submit the following items in a zipped folder:

- A .pdf document with details on the members of the group, names and student numbers. Here you may also give some information on the project.
- A well commented code that produces the expected output as outlined in the discussion above.
- The output of the program run using the input file (available on the moodle page):
  - `triangle_triples.data`
- Report a validation of your code for the triple $(3, 4, 5)$ using (1). That is report the exact value and the approximate value for a series of different N, you may need to chose N quite large.

- Include both text (i.e. tables presenting the results discussed in the previous section) and graphical output (graphics can be saved using the buttons on the bottom of the Figure window, or using the commands exemplified in the program `matplotexample.py` available on the moodle page).