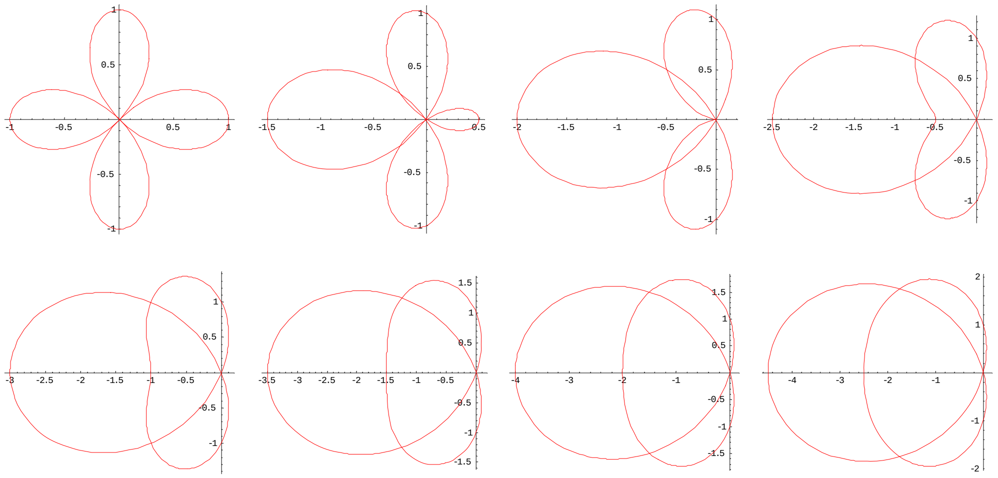
# Scarabaeus Curve



In the figure above are some examples of the *Scarabaeus Curve*, named after its likeness to the scarab beetle. This curve takes the polar form

$$r(\theta) = \cos(2\theta) - a\cos(\theta), \qquad \theta \in [-\pi, \pi]$$

where $\theta$ is the angle of the vector from the origin to the current curve point with the positive $x$-axis, $r$ is the distance between the current point and the origin, and $a$ is a parameter. The figure shows curves with $a$ running from 0 to 3.5.

For a little more information, take a look at
http://mathworld.wolfram.com/Scarabaeus.html

**Assignment 1**
Implement a utility that displays a Scarabaeus curve on the screen. It must be possible to set the (real) parameter $a$ interactively. You can approximate the curve by sampling $\theta$; make sure you sample dense enough to obtain a smooth curve.

**Assignment 2**
Implement a utility that can draw Cardinal splines, given a set of control points. Note: it is not necessary to make this interactive, except for a way to set the tension parameter *t*; this utility will be used for assignment 3.

**Assignment 3**
Implement a utility that samples points of the Scarabaeus curve uniformly (i.e. with fixed distance between $\theta$ samples) and consider these samples control points for a Cardinal spline curve. Draw the Cardinal spline. The goal here is to approximate the Scarabaeus curve closely using the spline curve.

**Question**
Describe a way to estimate the error between the Cardinal spline approximation and the actual Scarabaeus curve. This error will vary if you alter the tension parameter $t$ of the Cardinal spline.

**You must do this assignment with a partner.**[1]

You may use Java or C/C++ for your implementation. Use of 3rd party code for support tasks (GUI, drawing pixels or lines on the screen, etc.) is acceptable, just make sure that all of the spline code is your own.
If you use Java, a working webpage/applet[2] combination must be submitted. In the case of C/C++ a standalone PC executable must be submitted. **Submitting only** u**ncompiled source code is not acceptable.** In your submission, include all of your source code and a **concise** document detailing (1) your implementations and (2) the answer to the question. Pack everything in a zip/rar file and submit it using the 'submit' system.

The deadline is wednesday 10 March, 17:00.

[1] Working alone is only allowed if you (1) show that working with a partner is truly impossible in your case, and (2) you request an exception from the assistants before Monday, 1 March 2010.

[2] Submitting an **executable (!)** jar file is acceptable, but beware that we have had a lot of non-functional jar files submitted in the past. Creating an executable jar file that works well in a 'clean' Java environment is not entirely trivial.