Geometry
Second mid-term exam
5.5.2008

1. Let F_i be the reflection (in the plane) in the line ℓ_i, $i = 1, 2$. Denote by F the mapping $F_2 \circ F_1$. Show that the following hold:

   (a) If the lines ℓ_1 and ℓ_2 are parallel, then F is a translation.

   (b) If the lines ℓ_1 and ℓ_2 intersect at the point P, then F is a rotation around P.

2. Let Y be a circle in the plane, with center O and radius R, and let P be a point outside Y whose distance to O is d. Determine those values of the constant k for which the image of Y in the dilatation $P[k]$ (with center P and ratio k) is tangent to Y.

3. The *lemniscate of Gerono* (also known as the *figure eight curve*) is the plane curve formed by those points P', which are associated with points P of the unit circle $x^2 + y^2 = 1$ in the following way. We denote by Q the point of intersection of the horizontal line through P with the line $x = 1$, and further, we denote by ℓ the line through the point Q and the origin. Then P' is the point of intersection of ℓ and the vertical line through P.

   (a) Show that the lemniscate of Gerono is the image of the unit circle under the mapping $f : \mathbb{R}^2 \to \mathbb{R}^2$, where $f(x, y) = (x, xy)$. Derive a parametric representation $t \mapsto (x(t), y(t))$, $t \in [0, 2\pi]$, for the lemniscate.

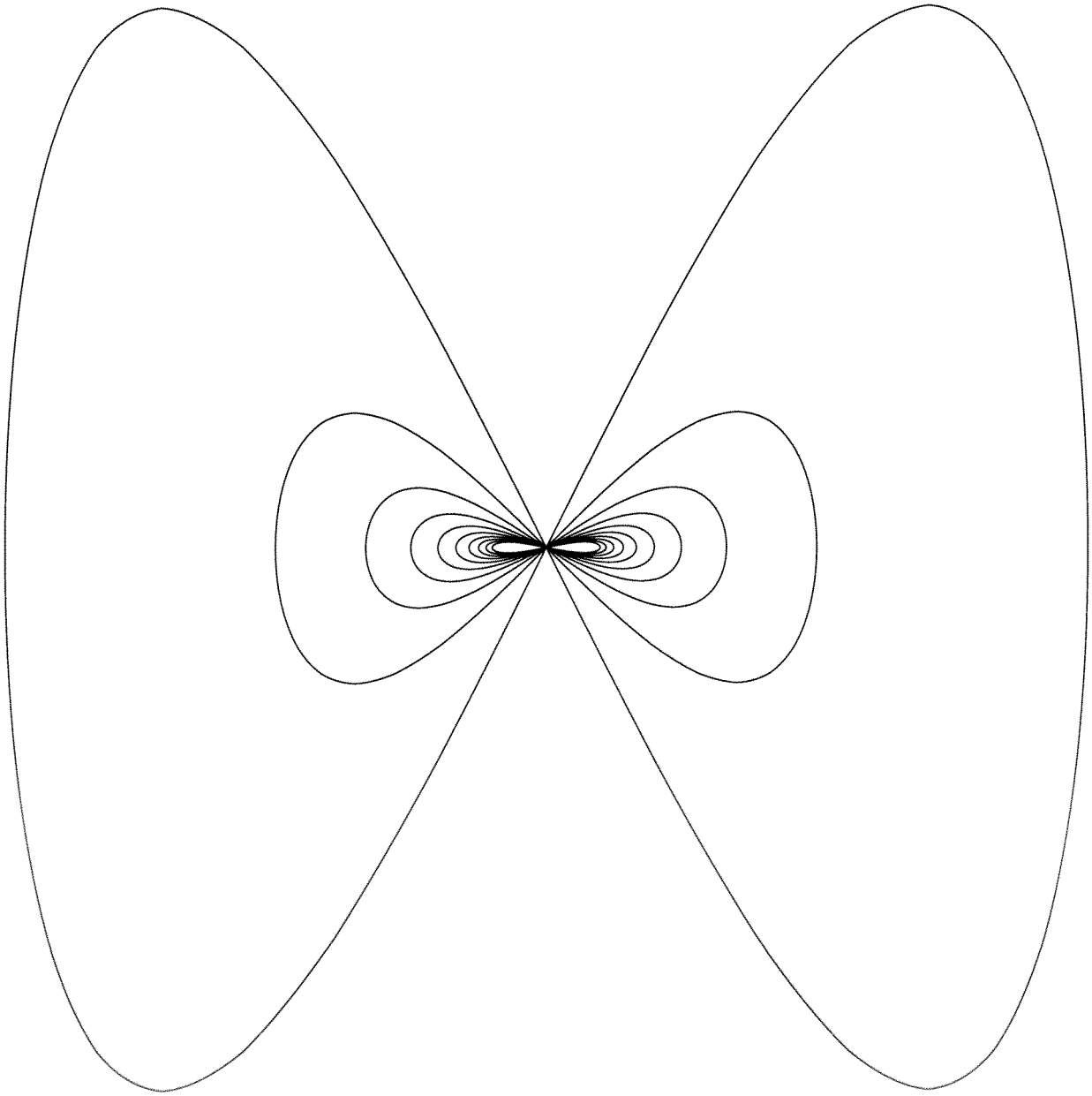
   (b) Let s and t be two numbers with $0 \leq s < t \leq 2\pi$. Set $x_0 = x(s)$, $y_0 = y(s)$, $x_3 = x(t)$ and $y_3 = y(t)$. Determine the control points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ in such a way that the PostScript command sequence "x_0 y_0 *moveto* x_1 y_1 x_2 y_2 x_3 y_3 *curveto*" determines a Bezier-curve whose tangents at the points $P_0 = (x_0, y_0)$ and $P_3 = (x_3, y_3)$ have the same direction as those of the lemniscate.

4. Write a PostScript routine f which replaces two numbers x and y, given at the top of the stack, by the numbers x and xy. Then write a PostScript program which starts with an assignment of a natural number, say 10, as a value for a variable k. The purpose of the program is to draw a pencil of lemniscates similar to that in the figure drawn on the other side of the paper. The curves in the figure are images of circles of radii $\frac{2}{i}$, for $i = 1, ..., k$, centered at the origin, under the mapping $f : (x, y) \mapsto (x, xy)$. In your program, you can use the PostScript routines *ctransform* and *subdivide* described in Casselman's book. The name $/f$ of the routine f is given (at the top of the stack) as a parameter for *ctransform*, and *ctransform* then transforms the prevailing current path, representing a circle (or a pencil of circles), into a path representing a lemniscate (or a pencil of lemniscates).
   (To give the right shape for the images of circles, the current path should be chopped into pieces with the commands *subdivide subdivide subdivide* before the execution of *ctransform*.)

[*Hint for Problem 1*: The classification of plane isometries. *Hint for part (b)*: For every point P, every point of the line ℓ_i is equidistant from the points P and $F_i(P)$.]

[*Hint for the first part of 3(a)*: Draw a picture of the situation and determine the y-coordinate of P' by using similar triangles or equations of lines. *Hint for the second part*: Derive the parametric representation with the help of the result of the first part and the standard parametric representation of the unit circle.]

[*A reminder for Problem 4*: Make sure that no extra line-segments are produced in connection with the command *arc*.]



Summary of some PostScript commands

The symbol $\emptyset$ means no arguments, or no return value.

1. Mathematical functions

Arguments	Command	Left on stack; side effects
$x\ y$	`add`	$x + y$
$x\ y$	`sub`	$x - y$
$x\ y$	`mul`	xy
$x\ y$	`div`	x/y
$x\ y$	`idiv`	the integral part of x/y
$x\ y$	`mod`	the remainder of x after division by y
x	`abs`	the absolute value of x
x	`neg`	$-x$
x	`ceiling`	the integer just above x
x	`floor`	the integer just below x
x	`round`	x rounded to nearest integer
x	`truncate`	x with fractional part chopped off
x	`sqrt`	square root of x
$x\ y$	`atan`	the polar argument of the point (x, y)
x	`cos`	$\cos x$ (x in degrees)
x	`sin`	$\sin x$ (x in degrees)
$x\ y$	`exp`	x^y
x	`ln`	$\ln x$
x	`log`	$\log x$ (base 10)

2. Stack operations

x	`pop`	$\emptyset$
$x\ y$	`exch`	$y\ x$
x	`dup`	$x\ x$
$x_0 \ldots x_{n-1}\ n\ i$	`roll`	$x_i \ldots x_0 \ldots x_{i-1}\ ni$

3. Dictionaries

name item	`def`	makes an entry in the current dictionary
n	`dict`	puts a dictionary of n null entries on the stack
dictionary d	`begin`	opens d for use
	`end`	closes the last dictionary opened

4. Conditionals

The first few return 'boolean' constants `true` or `false`. A few others have boolean values as arguments.

$x\ y$	`eq`	$x = y$?
$x\ y$	`ne`	$x \neq y$?
$x\ y$	`ge`	$x \geq y$?
$x\ y$	`gt`	$x > y$?
$x\ y$	`ge`	$x \geq y$?
$x\ y$	`gt`	$x > y$?
$x\ y$	`le`	$x \leq y$?
$x\ y$	`lt`	$x < y$?
$s\ t$	`and`	s and t are both true?
$s\ t$	`or`	at least one of s and t is true?
s	`not`	s is not true?
$s\ \{\ldots\}$	`if`	executes the procedure if s is true
$s\ \{\ldots\}\{\ldots\}$	`ifelse`	executes the first procedure if s is true, otherwise the second

5. Arrays

an array a	`length`	number of items in the array a
$a\ i$	`get`	a_i
$a\ i\ j$	`getinterval`	$a_i \ldots a_j$
a	`aload`	$a_0 \ldots a_{\ell-1}\ a$ (ℓ is the length of a)

6. Loops

$i\ h\ f\ \{\ldots\}$	`for`	steps through the loop from i to f, incrementing by h
$n\ \{\ldots\}$	`repeat`	executes the procedure n times
$\{\ldots\}$	`loop`	executes the procedure until `exit` is called from within the procedure
$\emptyset$	`exit`	exits the loop it is contained in

7. Graphics state

$\emptyset$	`gsave`	saves the current graphics state, installs a new copy of it
$\emptyset$	`grestore`	brings back the last graphics state
x	`setlinewidth`	sets current linewidth to x (in current units)
x	`setlinecap`	determines how lines are capped
x	`setlinejoin`	determines how lines are joined
$[\ldots]\ x$	`setdash`	sets current dash pattern
g	`setgray`	sets current colour to a shade of grey
$r\ g\ b$	`setrgbcolor`	sets current colour

8. Coordinates

Here, a matrix is an array of 6 numbers. The CTM is the **C**urrent **T**ransformation **M**atrix.

$\emptyset$	`matrix`	puts a matrix on the stack
matrix m	`defaultmatrix`	fills m with the default TM, leaves it on the stack
m	`currentmatrix`	fills the matrix with the current CTM, leaves it
$x\ y$	`translate`	translates the origin by $[x, y]$
$a\ b$	`scale`	scales x by a, y by b
m	`concat`	multiplies the CTM by m
$m_1\ m_2\ m_3$	`concatmatrix`	fills m_3 with the matrix product $m_1 m_2$
$x\ y$	`transform`	$x'\ y'$, transform of $x\ y$ by the CTM
$x\ y\ m$	`transform`	$x'\ y'$, transform of $x\ y$ by m
$x\ y$	`itransform`	$x'\ y'$, transform of $x\ y$ by the inverse of the CTM
$x\ y\ m$	`itransform`	$x'\ y'$, transform of $x\ y$ by the inverse of m
$m_1\ m_2$	`invertmatrix`	m_2 (the matrix m_2 is filled by the inverse of m_1)

9. Drawing

$\emptyset$	`newpath`	starts a new path, deleting the old one
$\emptyset$	`currentpoint`	the current point $x\ y$ in device coordinates
$x\ y$	`moveto`	begins a new piece of the current path
$x\ y$	`lineto`	adds a line to the current path
$x\ y$	`rmoveto`	relative move
$x\ y$	`rlineto`	relative line
$x\ y\ r\ a\ b$	`arc`	adds an arc from a to b, centred at (x, y), of radius r
$x\ y\ r\ a\ b$	`arcn`	negative direction
$x_1\ y_1\ x_2\ y_2\ x_3\ y_3$	`curveto`	adds a Bezier curve to the current path
$\emptyset$	`closepath`	closes up the current path back to the last point moved to
$\emptyset$	`stroke`	draws the current path
$\emptyset$	`fill`	fills the outline made by the current path
$\emptyset$	`clip`	clips drawing to the region outlined by the current path