Introduction to Mathematical Software
Exercise 3-5

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Advice

Visit the course website regularly to see whether there were updates to this exercise. This is version 1.

Problem 1 Lists and Sets in Maple

a) Write a Maple procedure that lists all factorials that are less than a given natural number. Example: input: 7, output: [1,2,6]. *Hint:* Use the lecture notes and the Maple help to find out about procedures in Maple (Keyword: `proc`).

b) Write a Maple procedure that returns for a given set M and a given number k all subsets that do *not* have exactly k elements. *Hint:* Have a look at the `combinat` package, especially how to utilize either the command `subsets` or `powerset`.

**Problem 2 Decimal Expansion of Rational Numbers**

Let two integers a, b be given with $b \neq 0$. Write a Maple procedure that returns the rational number $\frac{a}{b}$ in *decimal expansion*.

The output might be finite or periodic and should be of the form `"1.0"` for $a = 1$, $b = 1$ and `"-0.75"` for $a = -3$, $b = 4$ in the finite case and `"0.p3"` for $a = 1$, $b = 3$, `"0.1p6"` for $a = 1$, $b = 6$ and `"-0.00p142857"` for $a = -1$, $b = 700$, where p denotes the beginning of the period.

Browse the internet for a suitable algorithm to accomplish this task. Search engines should lead you to pages like http://en.wikipedia.org/wiki/Decimal.

Hint: Helpful Maple-commands may include `Array(1..n)` (where n is fixed), `cat`, and `irem`.

Don't Drink and Derive – Episode 2
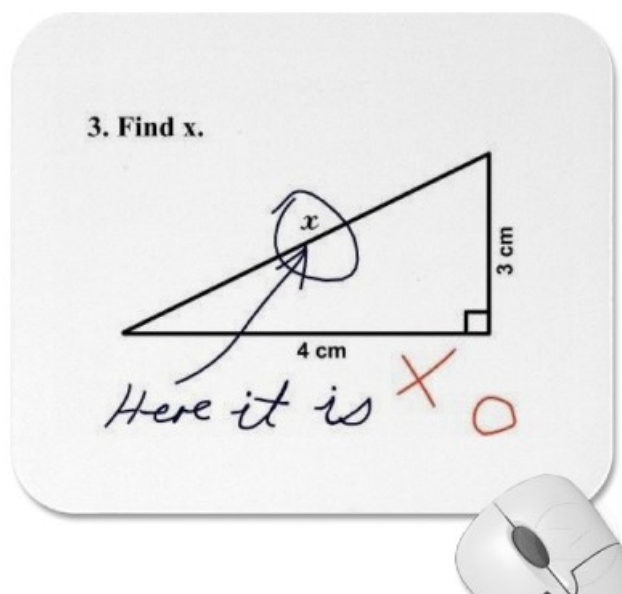
Problem 3 Design of a Beer Glass

Notice: If not stated otherwise, you are supposed to solve the problems analytically.

A brewery would like to design their own beer glass. It is your job to help them design the glass by using solids of revolution.

More information on solids of revolution can be found at http://en.wikipedia.org/wiki/Solid_of_revolution.

The solid of revolution around the x-axis is supposed to be given in terms of a function $p(x)$. A designer has given the following requirements (with lengths in cm):

$$p(0) = \frac{11}{4} \qquad p(6) = \frac{19}{10} \qquad p(18) = \frac{9}{2} \qquad p(20) = 4 \qquad p'(6) = p'(18) = 0$$

Although polynomial interpolation is often not a very good approach (interpolating large point sets can lead to strong oscillations and spline interpolation should be used instead), it is sufficient for what we want to achieve here. As you know, a polynomial of order n is uniquely defined by $n+1$ function constraints (which might be function or derivative values). As we are given 6 values, we are looking for a polynomial of degree 5.

a) Find the polynomial $p(x)$ of degree 5 that satisfies the specification above.

b) The glass should have a height of 20cm. Plot the function $p(x)$ (not the solid of revolution) on the interval $[0, 20]$. Make sure that both axes are scaled identically.

Next, we would like to visualize the solid of revolution. To this end, the Maple method `VolumeOfRevolution` is helpful. Before you continue, find out which package you have to include and what the parameter `output` does.

c) Plot the solid of revolution on the interval $[0, 20]$. *Hint:* Store the plot in a variable for the upcoming tasks and draw it using `display` from the `plots` package. There are faster/easier ways than to use `3dplot`.

The glass is to be calibrated to 500ml.

d) Determine the x-position of the calibration mark. You can do this numerically as the calibration mark cannot be positioned exactly anyway due to its thickness.

*) Plot the calibration mark and the bottom of the glass. *Hint:* the Maple command `spacecurve` might be useful.

You want to hide a cylinder with height $(20 - L)$ inside the glass. In particular, you would like to know the volume of the largest cylinder you can put the glass over (which need not be the largest cylinder fitting inside the glass). This means that the cylinder ranges from L to 20 in x-direction. (cf. figure 1)

e) Find $L \in [0, 20]$, such that the volume of the cylinder fitting under the glass becomes maximal. You can compute the extremal values using `fsolve`, if necessary.

f) Visualize the glass with the inscribed cylinder.

g) Compute the volume of the cylinder. (It should be larger than 235).

Don't Drink and Derive – Episode 3

Q: What is the difference between a mathematician and a philosopher?
A: The mathematician only needs paper, pencil, and a trash bin for his work – the philosopher can do without the trash bin...
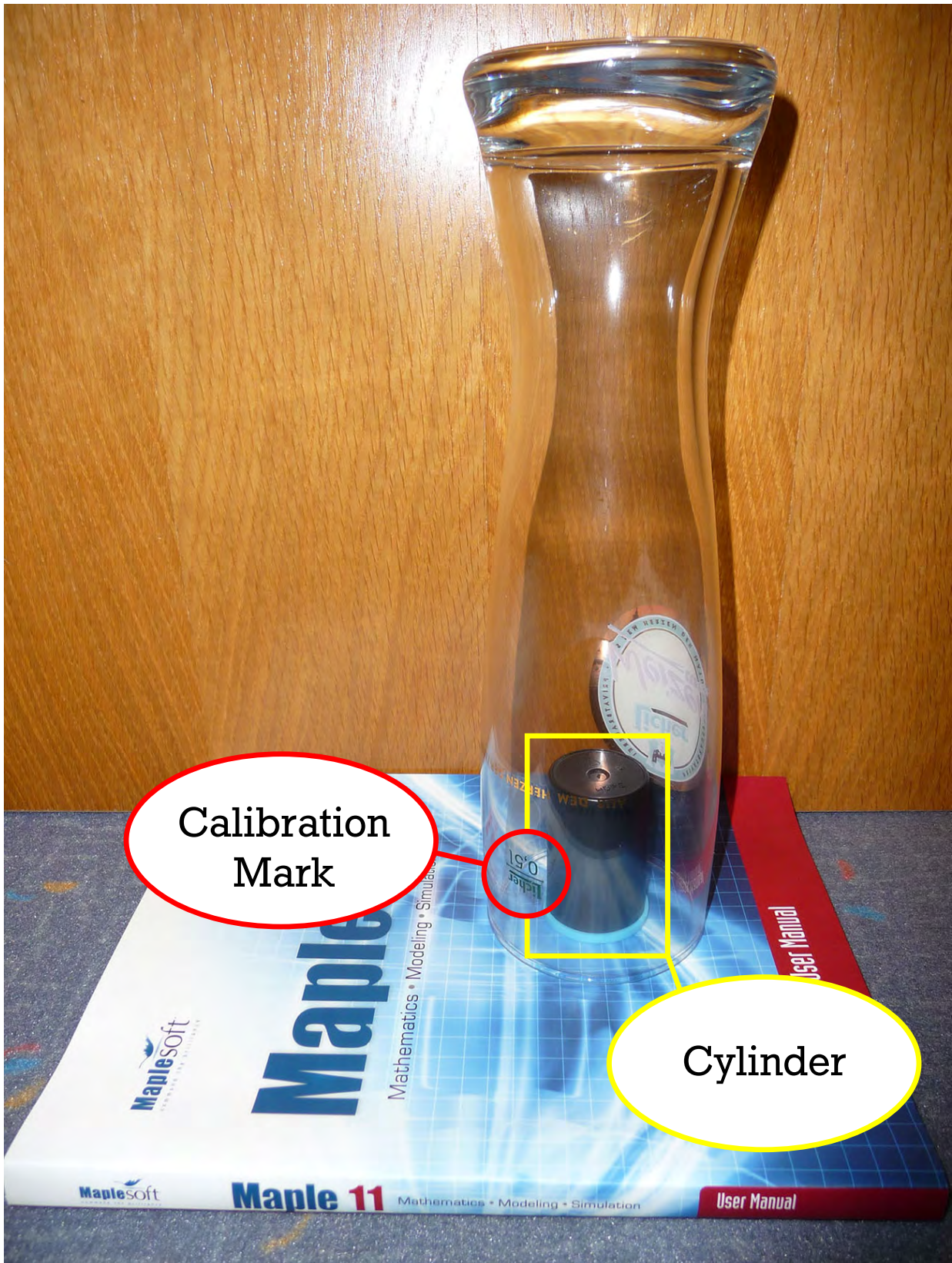
Figure 1: Beer Glass with Calibration Mark put over a not yet maximal Cylinder

Problem 4 An Application: Image Processing

Important: Before you start solving this exercise, ensure that you have solved all previous exercises!
Important: Load the package `ImageTools`. You will need some of its commands for this exercise.
Hint: For debugging-purposes, the commands `View`, `Preview` and `Write` might be useful. Check out what they do!

   a) Go to the course website. Download `Image1.jpg` from the "Exercises"-section. Remember the full path where you save it (e.g. `/home/username/Image1.jpg`).

   b) Load the Image into a variable using the `Read`-command.
      (*Hint for Windows-users:* Use this notation for paths: `C:/path/to/my/file/Image1.jpg`)

   c) Convert the image to grayscale using the `ToGrayscale`-command. Now you have a 2D-array of floating point numbers between 0 and 1. Each element of the array represents one pixel. 0 means black, 1 means white. If your array is `A`, you can access the pixels by `A[row,column]`, e.g. `A[10,10]`.

   d) Rotate this image by 90° counterclockwise using the `Rotate`-command. Take a look at the result.

   e) Create a new image with the same dimensions as the rotated image. Write a (nested) loop that sets all pixels to 0.5 (gray). *Hint:* Helpful commands: `Create`, `Height`, `Width`.

   f) We want to reduce the noise in our image using the so-called *median-filter*. Create a new image with the same dimensions as the rotated image. Do not modify the pixels of the original image! In the new Image, we set all inner pixels to the *median* of the original pixel and the 8 pixels around it. (This may take some seconds.) You can calculate the median by applying the `Statistics[Median]`-command to a list of these 9 pixels. Of course this does not work for the border-pixels of the image. For this exercise, it is perfectly ok to ignore them completely. Again, take a look at the result to see whether the filter worked as intended.

   g) Now we want to detect edges in your filtered image (if your are interested in details and further information, have a look at http://en.wikipedia.org/wiki/Edge_detection). For this purpose, we will use 3×3-filters. They tell us how to calculate the value of the new pixel from the old pixel and the 8 pixels around it by weighting them. Fortunately, this is already implemented in Maple (`Convolution`-command). Apply the following *Sobel-filters* (they are kind of an approximation of the derivative) separately to the image and save the results into different variables:

$$\text{SobelX} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad\qquad \text{SobelY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

   They detect vertical and horizontal edges. (cf. http://en.wikipedia.org/wiki/Sobel_operator)

   h) Unfortunately, the values of some pixels might become less than 0. These will become black, if you write the image directly, although they represent edges. Also, we only get one image with vertical edges and one with horizontal edges. We can avoid both problems by "defining" the *Sobel-Operator* as follows: If GX is the result of the `Convolution`-command when applying the SobelX-filter and GY is the result of the `Convolution`-command when applying the SobelY-filter, the result of the Sobel-operator is the result of the following Maple-Command:

$$GS := \text{sqrt} \sim \left(GX^2 + GY^2 \right); \hspace{4cm} \text{(Maple 14)}$$

$$GS := convert(map(sqrt, GX^2 + GY^2), Array, order = C\_order, datatype = float_8); \hspace{1cm} \text{(Maple 11)}$$

   Take a look at GS. The Edges should be white while the rest should be dark.

---

* "Advanced" Image Processing

If you still have a lot of time, try to solve one of the following exercises. We will not create any solutions for these exercises. If you solve one of the exercises, feel free to send us your solution. We might add it to the "Solutions"-section of the course website.

   a) Find a way to apply the median-filter to the border pixels. (There are several ways.)

   b) Rotate the image by an arbitrary angle.

   c) Find out how colored images work and apply the median-filter to them.

   d) Compare the Sobel-filters to Prewitt-filters, Roberts-filters and Laplace-filters.