Winfried Auzinger
Kevin Sturm

Sommersemester 2019
13.03.2019

Übungsaufgaben zur VU Computermathematik

Serie 1

Generelle Anmerkungen zu den Maple-Übungen:

- Die Aufgabenstellungen sind in englischer Sprache formuliert. Nehmen Sie sich die Zeit, diese genau durchzulesen. Manche Angaben enthalten Hintergrundinformationen über das jeweilige Thema und können daher gelegentlich etwas ausführlicher geraten. Was *konkret zu tun* ist, ist dann der Deutlichkeit halber in *kursiver Schrift* formuliert.
- Viele Übungsaufgaben sind keine reinen 'Maple-Aufgaben', sondern beinhalten eine mathematische Problemstellung, die Sie, ggf. mit entsprechenden Hinweisen, zunächst verstehen bzw. knacken sollen. Manche andere wieder sind experimenteller Natur. (Was für den Physiker das Labor ist, ist für den Mathematiker der Computer.)

  Bedenken Sie: Der Name unserer LVA ist Computer **Mathematik**.
- Manche Details müssen Sie ggf. noch selbst herausfinden. Orientieren Sie sich auch mit Hilfe des Flyers (siehe Homepage). Einige der Aufgaben haben auch den Zweck, dass Sie sich einen in der Vorlesung (aus Zeitgründen) nicht oder noch nicht im Detail besprochenen Stoff aktiv anhand von Beispielen selbst erarbeiten, z.B. was die Erstellung von Grafiken, Animationen etc. betrifft.
- Der Lösungsweg ist nicht immer eindeutig; gegen kreative Alternativlösungen ist nichts einzuwenden.
- Eine allfällige Formulierung '*verify by examples ...*' bedeutet, eine Aussage anhand konkreter Beispiele zu überprüfen; dies bedeutet natürlich nicht Verifikation in einem. strengen Sinn.
- Für vielen Fragestellungen gibt es innerhalb von Maple schon fertige Lösungen. Es spricht aber nichts dagegen, so etwas als Übungsaufgabe zu verwenden. (Auch in anderen Übungen berechnen oder beweisen Sie Dinge, die schon andere vor Ihnen berechnet bzw. bewiesen haben.)
- Ein Hinweis der Form `?` *command* bedeutet: Konsultieren Sie die Maple-Hilfe zu *command*.
- Nützen Sie generell die Maple-Hilfe systematisch − für die praktische Arbeit ist dies unumgänglich. Es sei auch darauf hingewiesen, dass das Verhalten eines derartigen Systems nicht immer genau vorhersehbar ist und man daher manches durch Ausprobieren herauszufinden versuchen wird. (Auch von Version zu Version kann sich das Verhalten manchmal ändern.)
- Ihre Codes sollten Sie so weit wie möglich anhand von Beispielen testen. Bei der Vorführung im Computerlabor sollen Sie in dieser Weise die Korrektheit Ihrer Ausarbeitungen dokumentieren.
- Dokumentieren Sie Ihre Worksheets auch in angemessener Weise mittels Zwischentexten bzw. Kommentaren (`#`). Das wesentliche Ziel dabei sollte immer sein, dass Sie selbst später noch erkennen können, was Sie sich dabei gedacht haben.
- Mit (∗) markierte Aufgaben (kommt manchmal vor) sind ein wenig anspruchsvoller. Sie können positiv gewertet werden, wenn Sie sich seriös damit befassen, auch wenn Sie keine vollständige Lösung vorweisen können.

Exercise 1.1: ***Playing with lists; playing with prime numbers.***

a) *Design a function* `primes(n)` *which returns a list of the first n prime numbers.*

Hint: use `?ithprime`. Also consult Maple Help to understand the difference between `ithprime` and `nextprime`.[1]

b) Consider numbers of the form $F_n = 2^{2^n} + 1$ with $n \in \mathbb{N}$, and *find the smallest $n \in \mathbb{N}$ for which F_n is* not *a prime number. What are the prime factors of that number?*

c) A list `L` is called palindromic if `L[`i`]=L[`$n{+}1{-}i$`]` for $i = 1 \ldots n$, where n denotes the length of `L`.

Design a function `ispalindromic(L)` *which returns* `true` *if* `L` *is palindromic, and* `false` *otherwise.*

Exercise 1.2: ***Playing with (finite) sets.***

a) *Design a function* `arithmetic_mean(S)` *which assumes that the argument* `S` *is a set of numbers, and which returns the arithmetic mean of these numbers.*

Remark: The 'numbers' need not to be numerically specified. Try different examples.

b) *Check the Maple operator* `?union`*, and design a function* `u2(.,.)` *which behaves in the same way (returning the union of two given sets).*

c) *Design a function* `u3(.,.,.)` *which returns the union of three given sets.*

d) *Design a function* `i4(.,.,.,.)` *which returns the intersection of four given sets.*

Exercise 1.3: ***Euler, schau owa, (i).***[2]

a) Euler's constant $\mathrm{e} = 2.718\ldots$ is defined by

$$\mathrm{e} = \lim_{n\to\infty} e_n, \quad \text{with } e_n = \Big(1 + \frac{1}{n}\Big)^n .$$

Find out experimentally how large n has to be chosen such that the rational approximation e_n to e satisfies $|e_n - \mathrm{e}| < 10^{-4}$.

b) Alternatively,

$$\mathrm{e} = \lim_{n\to\infty} s_n, \quad \text{with } s_n = \sum_{k=0}^{n} \frac{1}{k!} .$$

Again, as in **a)***, find out experimentally how large n has to be chosen such that the rational approximation s_n to e satisfies $|s_n - \mathrm{e}| < 10^{-4}$.*

c) Same question as in **b)**, but for

$$\mathrm{e}^x = \lim_{n\to\infty} s_n(x), \quad \text{with } s_n(x) = \sum_{k=0}^{n} \frac{x^k}{k!} \quad \text{and } x = 10,\ x = 100.$$

Be careful: It will be necessary to choose a higher numerical precision using `evalf` to obtain reasonable results. (The default value for the relevant system variable `Digits` is `10`, which is rather small.)

Exercise 1.4: ***A technique for symbolic summation.***

<u>Partial summation</u> is a discrete analogue of partial integration: For $(a_k), (b_k)$, $k = 0 \ldots n$ $(n \in \mathbb{N})$ we have[3]

$$\sum_{k=1}^{n} a_k (b_k - b_{k-1}) = a_n\, b_n - a_0\, b_0 - \sum_{k=1}^{n} (a_k - a_{k-1})\, b_{k-1} .$$

a) *Use this technique to derive an explicit formula for the generalized geometric sum*

$$\sum_{k=1}^{n} k\, q^k .$$

Hint: Choose $a_k = k$, evaluate the resulting formula by hand, and enter it as a Maple expression. Compare with the result delivered by direct use of `sum`. Also check the outcome for some numerical values of n.

[1] These are nontrivial number-theoretical functions.

[2] The underlying analysis is not really difficult, but here we simply confine ourselves to experimental observation.

[3] The proof of this formula is easy: Combine the sums and simplify – this yields a telescoping sum.

b) (*) *Now, consider*

$$\sum_{k=1}^{n} k^p \, q^k \,,$$

where p is declared to be an arbitrary positive integer via `assume(p,posint)`.

Attention: What is the problem here? Do not struggle too much with this, but try to understand that it is rather tricky. And *test:*

– *Does direct use of* `sum` *give you an explicit answer, if a numerical value for $p \in \mathbb{N}$ is not explicitly specified?*

– On the other hand, for particular numerical values $p = 1, 2, 3, \ldots$, `sum` easily finds the result (*check*).

Exercise 1.5: ***Plotting.***

a) Use `plot` to plot some real functions of your choice. Check the Maple help and try to generate a 'nice' plot by adjusting options. E.g., modifying the default values of the options `color`, `thickness`, `axes` (and some others) may be useful.

b) In Exercise 1.4 we have considered sequences of the form $(a_k) = (k^p \, q^k)$ $(k = 0, 1, 2, \ldots)$.

Use[4] `?plots[pointplot]` *to visualize the behavior of these sequences, e.g. for $q = \frac{1}{2}$ and $p = 0, 1, 2, , 3 \ldots$.*

c) Plotting commands generate a special data structure and immediately render the result. But sometimes it is useful to combine several plots in a single picture. To this end, the individual plot structures are saved (by assigning names to them, like for instance `p1:=plot(...):`, `p2:=pointplot(...):` and then rendering them together in a single picture using `?plots[display](p1,p2,...)`

Check and test. (You are happily invited to choose options influencing the optical outcome.)

Exercise 1.6: ***Euler, schau owa, (ii).***

Consider the sequence (H_n), defined by

$$H_n = \sum_{k=1}^{n} \frac{1}{k} \qquad (n \in \mathbb{N}).$$

a) (Probably you already know the answer:) *Is there an elementary algebraic formula for the values H_n in terms of n?* (Check the outcome of `sum`.)

b) *Use* `?plots[pointplot]` *to visualize the evolution of the values (H_n) for $n = 1, 2, 3, 4, 5, \ldots 10, 11, 12, \ldots 100, 101, 102, \ldots$ and so on. From this plot, can you draw any conclusion about its behavior for $n \to \infty$?*

c) There are efficiency issues here:

- First of all, computing the sums H_n is much more efficient in floating point arithmetic than in exact rational arithmetic (*explain*).
- Moreover, if you use a function of the form `n->add(1/k,k=1..n)` (rather, its analog in floating point arithmetic) evaluating the sum H_n in a straightforward way: *Is this computationally efficient for* **b)** *?*

  Suggest a reasonable alternative (informally, at this stage not necessarily in terms of syntactical details specific to Maple – this is not difficult bot not the topic of this exercise).

d) From **b)** you may conclude that the n-axis should be scaled logarithmically. *Do it* (choose base 10).

e) You probably know what is the 'limit' of H_n for $n \to \infty$.

Now repeat **b)***, plotting $(H_n - \ln n)$ instead of H_n. What do you observe?*

Exercise 1.7: ***A formula attributed to Ramanujan.***

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4\,k)! \cdot (1103 + 26390\,k)}{(k!)^4 \cdot 396^{4k}}$$

a) *Use the first 12 terms of this series to compute an approximation for π and check how accurate this appriximation is.*

Hint: Use `evalf` and adjust the value of `Digits` until you are sure that your result is O.K.

[4] `pointplot` is a function from the package `plots`.

b) *Design a nice plot visualizing the convergence behavior of this series.*

Hint: The values (approximation errors) should be displayed on a logarithmic ($\log_{10}$) scale.

Exercise 1.8: ***Decimal digits.***

a) *Design a function* `digit(n,k)` which expects a natural number `n` $\in \mathbb{N}$ as argument and returns its `k`-th decimal digit. E.g.,

```
digit(1034,1)  -->   4
digit(1034,2)  -->   3
digit(1034,3)  -->   0
digit(1034,4)  -->   1
digit(1034,5)  -->   0
...
```

Hint: Use `mod`.

b) *Design a function* `length(n)` which expects a natural number `n` $\in \mathbb{N}$ as argument and returns its 'length', i.e., the number of decimal digits of `n`. E.g., `1034` has 'length' 4.

Hint: Use the decimal logarithm `log[10](.)` and the rounding function `floor(.)` (rounding downwards to nearest integer).

c) *Use* **a)** *and* **b)** *to design a function* `sumdigits(n)` which expects a natural number `n` $\in \mathbb{N}$ as argument and returns the sum of its decimal digits.

Make sure that special cases like 1, 10, 100, 1000, ... are treated correctly.