Winfried Auzinger
Dirk Praetorius

Sommersemester 2018
25. April 2018

Übungsaufgaben zur VU Computermathematik
Serie 5

Generelle Anmerkung zu den Maple-Übungen:

- Die Aufgabenstellungen sind in englischer Sprache formuliert. Nehmen Sie sich die Zeit, die Aufgabenstellungen 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.
- Die meisten 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**.
- Lesen Sie die Angaben und Hinweise genau durch; manches müssen Sie ggf. noch selbst herausfinden. Orientieren Sie sich auch mit Hilfe des Flyers (siehe Homepage). Manche 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 (ggf. unter Umgehung der in einem Hinweis nahegelegten Vorgangsweise) ist nichts einzuwenden.
- Die Formulierung '*verify by examples ...*' bedeutet, eine Aussage anhand konkreter Beispiele zu überprüfen; dies bedeutet natürlich nicht Verifikation im 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 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 gewertet werden, wenn Sie sich seriös damit befassen, auch wenn Sie keine vollständige Lösung vorzuweisen haben.

Exercise 5.1: ***Playing with prime numbers (primes) and Euclid's proof.***

a) A very naive 'algorithm' for generating a sequence of primes would be based on the following idea inspired by Euclid's proof of the existence of infinitely many primes:

- Take the first prime, $p_1 = 2$.
- $p_2 := p_1 + 1 = 3$ is prime. (O.K.)
- $p_3 := p_1 \cdot p_2 + 1 = 2 \cdot 3 + 1 = 7$ is prime. (O.K.)
- ...

Check when this algorithm breaks down with a wrong answer. Explain why.

(This is easy to do by hand. Just for practice, do it in Maple, using `? ifactor` or `? isprime`.)

b) A professional implementation yielding the i-th prime is the Maple function `ithprime`.

Use `ithprime` *to design a function* `primes(n)` *which generates, for given* n*, a list of the first* n *primes.*

Hint: Use `seq`.

c) Assume that you already know, for given n, the correct sequence $p_1, \ldots, p_n$ of the first n primes (as not delivered by **a)** but by **b)**). Then one may presume that the next prime following p_n is given by

$$p_{n+1} = p_1 \cdot p_2 \cdots p_n + 1$$

This also is naive thinking. Determine the smallest value n *for which* p_{n+1} *defined in this way is not prime. Test and explain.*

d) *Was* A.D. 2017 *a 'prime year'? What is the next prime year following* 2018*?*

Exercise 5.2: ***Playing with the definition of*** e^x***. Some plots.***

The exponential function $e^x = \exp(x)$ can be defined in two equivalent ways:[1]

(i) $e^x := \sum_{k=0}^{\infty} \frac{x^k}{k!} = \lim_{n\to\infty} \sum_{k=0}^{n} \frac{x^k}{k!}$ (ii) $e^x := \lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n$

a) For given $n \in \mathbb{N}$, we consider the polynomials[2] of degree n,

(i) $p_n(x) := \sum_{k=0}^{n} \frac{1}{k!} x^k$ and (ii) $q_n(x) := \left(1 + \frac{x}{n}\right)^n = \sum_{k=0}^{n} \frac{1}{n^k} \binom{n}{k} x^k$

Design two functions `p(.,.)` *and* `q(.,.)` *which return* $p_n(x)$ *resp.* $q_n(x)$ *for given* x *and* n*.* (Use `add`, `expand`.)

b) *Use*[3] `? plot` *to display (for several values of* n*, e.g.,* $n = 5, 10, 15, \ldots$ *and for* $x \in [-1, 1]$*) the functions* e^x*,* $p_n(x)$*, and* $q_n(x)$ *in a single picture. What do you observe: Which polynomial (*$p_n(x)$ *or* $q_n(x)$*) is a better approximation for* e^x *?*

c) For $x = 1$, (i) and (ii) define two different sequences converging to $e = \exp(1)$.

Use `? plots[pointplot]`[4] *to visualize the convergence behavior of these sequences.*

Exercise 5.3: ***Playing with*** π***.***

There are many possible ways to approximate the number π (`Pi`) numerically, but a good (rapidly convergent) rational approximation is not so easy to find at an elementary level. A classical one is given by the Leibniz expansion

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots$$

This is nothing but evaluating the Taylor series of the arctan function,

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots$$

---

[1] The equivalence of (i) and (ii) is not trivial. This topic is treated in introductory analysis courses.

[2] p_n is the Taylor polynomial of degree n. Note that $p_n \neq q_n$; already the x^2-coefficients are different.

[3] `? plot/details` provides more detailed information.

[4] `pointplot` is used for plotting discrete data. It is contained in the package `plots`.

at $x = 1$. However, the Leibniz series converges very slowly. Based on properties of the arctan function, at the beginning of the 18th century John Machin found out that

$$\frac{\pi}{4} = 4 \arctan(\tfrac{1}{5}) - \arctan(\tfrac{1}{239}).$$

Combined with the Taylor expansion for arctan, this gives a series converging to $\frac{\pi}{4}$ much faster than the Leibniz series.

Play with both series (using `evalf`*) and compare their convergence behavior for increasing n. Document your findings in an appropriate way.*

Exercise 5.4: ***Taylor expansion of the*** arcsin ***function.***

The Taylor polynomial of degree $2n{+}1$ of the function $\arcsin x$ about $x_0 = 0$ is given by $(n = 0, 1, 2, 3, 4, \ldots)$

$$\arcsin x \;=\; \sum_{k=0}^{n} \binom{2k}{k} \frac{1}{4^k} \frac{x^{2k+1}}{2k+1} \;=\; \underbrace{x + \frac{1}{2}\frac{x^3}{3} + \frac{1\cdot 3}{2\cdot 4}\frac{x^5}{5} + \frac{1\cdot 3\cdot 5}{2\cdot 4\cdot 6}\frac{x^7}{7} + \ldots}_{n+1 \text{ terms}}$$

Here, $m!! = (m!)!$ denotes the double factorial.

a) *Design two functions which, for given n, generate these expressions (with exact rational coefficients). Verify that they are identical for $n = 0, 1, 2, 3, 4, \ldots$.*

Hint: Use `add` and `mul`.

b) For $x = 1$, the series converges to $\arcsin 1 = \frac{\pi}{2}$. *Is the convergence reasonably fast for increasing n ?*

(Use the second variant combined with `evalf`.)

c) Same question as in **b)**, for $x = \frac{\sqrt{2}}{2}$ with $\arcsin x = \frac{\pi}{4}$. [5]

Exercise 5.5: ***Some manipulations with lists.***

Let a list L of values be given. Use `seq`, `add`:

a) *Design a function* `rev(L)` *which returns the list L with the order of elements reversed.*

b) *Design two functions* `lodd(L)` *and* `leven(L)` *which returns a list containing the elements of L corresponding to odd and even positions, respectively.*

c) *Design a function* `cumsum(L)` *which returns the list* `[L[1], L[1]+L[2], L[1]+L[2]+L[3],...]`. [6]

d) *Design a function* `twins(L)` *which counts the number of twins in L, i.e., the number of pairs of successive elements which are identical.*

Hint: For instance, `[1,2,2,2,1,1]` contains 3 twins.

With `evalb(.=.)` you can compare two values. `evalb` returns `true` or `false`. For our purpose it is convenient to work with 1 and 0 instead, as in MATLAB. The following function converts `true` to 1 and any other value to 0:

```
t := b->ifelse(b=true,1,0)
```

(Here we have introduced the basic `ifelse` construct.) Using the function `t` you can realize **d)**.

Exercise 5.6: ***Some manipulations with sets.***

Assume a list L is given, where each list element is a set, e.g.,

```
[{1,2,3},{1,4},{2,3,3}]
```

a) *Design a function* `nadd(L)` *which returns the sum of all the sums of elements of the sets in L.*

Hint: Use `add`.

b) *Design a function* `nunion(L)` *which returns the union of all the sets in L in form of a set.* [7]

Hint: `op(.)` returns all elements of a set.

---

[5] Note that this involves evaluation of the irrational number $\sqrt{2}$.

[6] This could be realized in a more efficient way using a loop avoiding redundant computations, but we ignore this here.

[7] `union` only works for two sets.

c) *Design a function* `pwdiff(L)` *which returns* `true` *if the sets in L are pairwise different (i.e., if* `L[i] <> L[j]` *for all* $i \neq j$*), and* `false` *otherwise.*

Hint: Use `evalb`.

Exercise 5.7: ***A cubic equation.***

We define the polynomial function

```
p := x->x^3+b*x+c
```

with `b,c` unspecified (which means that these may be arbitrary complex numbers).

a) *Use* `solve` *to compute the exact formulas for the zeros of p.* This gives you three solutions ξ_1, ξ_2, ξ_3 (based on Cardano's formula, involving a third root [8]) which may be real or complex.

b) *Verify that these expressions* ξ_j *are indeed correct symbolic representations of the zeros of p by substituting them into p.*

c) *Generate the expression* $(x-\xi_1)(x-\xi_2)(x-\xi_3)$ *and verify that it is identical with* $p(x)$*.*

Hint: Use `simplify`.

Exercise 5.8: ***More polynomial equations.***

a) We consider the equation $x^n = 1$ with $n = 99$.

Use `solve` *to compute all solutions, and use* `? plots[complexplot]` *to visualize the location of these solutions in the complex plane.*

(Check the appropriate parameter settings for `complexplot` in order to produce a nice picture.)

Hint: You may first try $n = 2, 3, 4, \ldots$.

b) We consider the equation $x^5 = 1 + x$.

Check what happens if you try to solve this equation using `solve`.

Hint: A `RootOf` expression represents a solution which cannot be exactly computed.

c) Maple does a rather good job in numerically finding solutions to polynomial equations: If you apply `evalf` to the output delivered under **b)**, you get numerical approximations for all the roots (zeros) of the given equation.

Set `Digits` *to* 10 *and try this out.* You get 5 complex [9] numerical solution approximations ξ_i. *Use* `abs` *to compute the absolute values of the residuals* $\xi_i^5 - \xi_i - 1$ *of all these approximations.*

d) *Check what happens if you try to solve the equation from* **c)** *using the numerical root finder* `? fsolve`.

---

[8] Each real or complex number z has 3 third roots $\sqrt[3]{z}$ – in special cases 2 or 3 of them may coincide. $w = \sqrt[3]{z}$ means that $w^3 = z$ is valid.

[9] One of the solutions is real, and it is a priori clear that at least one real solution exists. Do you know why? (This question has nothing to do with Maple.)