Elementare Funktionen annähern – wie geht das?

H.R. Schneebeli

28. November 2022

Zusammenfassung

Die Analysis definiert einen abstrakten Funktionsbegriff. Diese Definitionen erlauben, wichtige Lehrsätze über allgemeine Eigenschaften ganzer Klassen von Funktionen zu beweisen. Ein typisches *Beispiel* ist der sogenannte *Zwischenwertsatz für stetige Funktionen.* Er garantiert für jede auf einem Intervall $I = [a, b] \subset \mathbb{R}$ mit $a < b$ definierte und stetige Funktion f, die *Existenz einer Lösung* in I zur Gleichung $f(x) = w$ für jedes beliebige w zwischen den Extremwerten von f auf dem Intervall $[a, b]$.

Das ist wohl, was Leibniz dachte, wenn er sagte: Natura non facit saltus – Die Natur macht keine Sprünge, weil er ihre Veränderungen als 'stetig' dachte.

Ferner begründet Analysis die *Existenz* wichtiger spezieller Funktionen, die sich durch besondere Eigenschaften auszeichnen, wie zum Beispiel Logarithmen oder andere *Standardfunktionen.* Dabei bleibt ein Problem offen: die konkrete Berechnung von Funktionswerten - oder realistischer, deren numerische Approximation für praktische Zwecke und Anwendungen.

Dazu dient die *numerische Analysis:* Sie entwickelt praktisch nutzbare Verfahren, um *Lösungen deren Existenz die Analysis abstrakt garantiert, numerisch konkret zu berechnen* oder – typischer – *effizient und hinreichend genau anzunähern.*

Zwei Beispiele für Aufgaben aus der Numerik sind:

1. *numerische Löser,* die zu Gleichungen $f(x) = 0$ mit einer berechenbaren, stetigen Funktion f gute Näherungslösungen bestimmen.
2. *Verfahren zur effizienten Berechnung* von guten Näherungen der Funktionswerte etwa bei den *'elementaren' Funktionen.*

Dieser Text zeigt aus der Sicht der gymnasialen Mathematik exemplarisch, wie sich *Quadratwurzeln, Exponentialfunktion und trigonometrische Funktionen* mit *numerischen Verfahren* annähern lassen. Das Thema ist geeignet für *Experimente* im Fach *Anwendungen der Mathematik* oder für die Begabtenförderung.

Die Entwicklung von *lokal* guten Näherungen wird ergänzt mit Massnahmen, welche lokale Näherungen auf einen grösseren Anwendungsbereich ausdehnen. Beim Übergang vom lokalen zum globalen Standpunkt werden oft besondere *algebraische Eigenschaften* der betrachteten Funktionen ausgenutzt. Beispiel: das *'Additionstheorem'* der Exponentialfunktion $\exp(x + y) = \exp(x) \cdot \exp(y)$.

Der Lernprozess wird in der Folge begleitet durch eine Reihe von *Aufgaben,* deren eigenständige Bearbeitung bis zur Lösung einen wesentlichen Beitrag zum Erwerb einer eigenen Fachkompetenz für alle Lernenden darstellt. Es ergibt sich so die Chance, in Beispielen erste Erfahrungen zu sammeln, *eigene Ideen* zu entwickeln, zu erproben und allenfalls zu verbessern. So lässt sich der Graben zwischen Theorie und Praxis oft überbrücken. Eine hinreichend gute Antwort soll genügen, Perfektion ist nicht das Ziel.

Professionell entwickelte numerische Algorithmen sind das Ergebnis von *numerical engineering,* stark bedingt durch Hard- und Software. Die gymnasiale Ausbildung kann dazu bloss *erste Erfahrungen und Einsichten* zum Thema vermitteln – und Respekt vor der Arbeit von Fachleuten.

1 Wo ist das Problem?

Die Analysis hat sich im Laufe des 19. Jahrhunderts ein Fundament gegeben mit der Charakterisierung der reellen Zahlen $\mathbb{R}$ und der komplexen Zahlen $\mathbb{C}$. Diese Zahlbereiche sind mit besonders guten Eigenschaften so ausgestattet, dass es gelingt, eine grosse Zahl von erwünschten Folgerungen zu ziehen und die kanonischen Lehrsätze der Analysis deduktiv herzuleiten. Zu diesen Folgerungen zählen allgemeine Aussagen über ganze Funktionenklassen, etwa alle stetigen Funktionen auf einem abgeschlossenen und beschränkten Intervall oder alle stetig differenzierbaren Funktionen. Dann lassen sich Aussagen beweisen wie der Zwischenwertsatz, der Satz vom Maximum, der Mittelwertsatz, der Hauptsatz der Integralrechnung sowie Aussagen zur Existenz oder Eindeutigkeit von Lösungen bei Differentialgleichungen.

Daneben gibt es aber 'spezielle Funktionen' mit einer bestimmten Rolle im Werkzeugkasten der praktischen Mathematik. Zum Beispiel: die Wurzelfunktion, trigonometrische Funktionen oder Logarithmen und Exponentialfunktionen. Solche Funktionen übernehmen Hilfsarbeiten etwa beim Lösen gewisser Gleichungen oder Differentialgleichungen. Sie sind wichtige Bausteine in ungezählten Anwendungen oder für die praktische Analysis. Nun zeigt sich, dass wir einen hohen Preis zahlen für die Sicherheit, die uns die Analysis mit ihren allgemeingültigen Sätzen garantiert. Wir können über die reellen Zahlen gut nachdenken, aber wir können nur beschränkt mit ihnen rechnen. Zum *numerischen Rechnen* mit dem Computer benutzen wir jeweils bloss einen *endlichen* Satz von *Maschinenzahlen* $\mathbb{M}_{\mathbb{R}} \subset \mathbb{Q}$, um die reellen Zahlen $\mathbb{R}$ so gut als möglich zu imitieren. Aber dies gelingt natürlich nur in sehr beschränktem Masse, und wir zahlen einen Preis: Diskretisierungsfehler und Rundungsfehler sind unvermeidbar.

Anderseits entziehen sich viele quantitative Probleme von praktischer Bedeutung einer formal exakten Behandlung im Rahmen der abstrakten *Analysis, die im Rahmen von* $\mathbb{R}$ *Einsichten sucht,* während *Numerik* in der Regel im Rahmen von $\mathbb{M}_{\mathbb{R}}$ *brauchbare Näherungen nur hinreichend genau aber hinreichend rasch anstrebt.*

In der Regel sind reelle Zahlen Idealbilder, die von Maschinenzahlen nur mehr oder weniger gut angenähert werden können. Die oft benutzte Anweisung: Wähle eine beliebige reelle Zahl zwischen im Intervall $I = [a, b]$, ist natürlich mit Maschinenzahlen nicht ausführbar, weil es nur endlich viele Maschinenzahlen gibt. Jede Manipulation mit Maschinenzahlen im Rechner ist ein physikalisches Experiment und damit eine potenzielle Fehlerquelle. Schon das Speichern einer Zahl x als Dezimalzahl verwischt deren Identität beim Runden auf eine Maschinenzahl. Jede arithmetische Grundoperationen mit Maschinenzahlen kann mit Rundungsfehlern behaftet sein. Noch viel drastischer wird es auf der Ebene der Funktionen.

Die Funktionen der Analysis sind sehr oft gedankliche Konstrukte, die sich beim praktischen Rechnen wie Phantome verhalten. Bestenfalls lassen sie sich durch gute Näherungen ersetzen. Diese Näherungen sind uns so vertraut, und sie sind meist so zuverlässig, dass wir sogar in der Fachsprache kaum je unterscheiden zwischen dem abstrakten Vorbild, etwa der Wurzelfunktion $\sqrt{\ }$, die zum Input $t > 0$ die einzige positive Lösung $\sqrt{t}$ der Gleichung $x^2 = t$ zurückgibt und einer Realisierung durch eine Näherung $\mathrm{sqrt}(t)$ auf einem konkreten Rechner. Hier zeigt sich das grundsätzliche Problem. Alle numerischen Antworten eines Computers sind rationale Zahlen, und sie sind in der Regel durch Rundungsfehler verrauscht. Schon $\sqrt{2}$ lässt sich in keiner Rechnerarithmetik exakt mit Dezimalzahlen ausgeben, denn diese Zahl ist irrational. Jede auf einem Computer realisierbare Näherungsfunktion sollte aus Sicht der Analysis eher gedacht werden als eine riesige Liste von einzelnen rationalen Näherungen zu Funktionswerten. Zwischen diesen klaffen – im Vergleich zu den Funktionen der Analysis – Lücken. An Stetigkeit, Zwischenwerteigenschaft oder Differenzierbarkeit ist im wörtlichen Sinne bei diesen Konstrukten nicht zu denken. Auch hat jede Rechnerfunktion globale Extrema, weil es nur endlich viele Maschinenzahlen gibt. Es zeigt sich so, dass Rechnerexperimente im allgemeinen kaum Aufschluss geben können über die Eigenschaften der Vorbilder dieser Funktionen aus der Analysis. Interessanterweise bietet die Numerik aber oft Ersatz und Ergänzungen in quantitativer Hinsicht an: Numerische Lösungen, numerische Ableitung, numerische Integration, Simulationen und angenäherte Lösungen von Differentialgleichungen sind aus der Anwenderpraxis nicht mehr wegzudenken. Wer sie mit Verstand anwendet, kann damit Schwierigkeiten überwinden, die uns die Analysis aufbürdet. Deren perfekte Ideale erscheinen als Phantome. Brauchbare und effiziente Näherungen sind in Anwendungen unverzichtbar.

Überblick

In den folgenden Abschnitten wird ein historischer Algorithmus zum angenäherten Berechnen von Quadratwurzeln benutzt, um einen einfacheren Fall vorzustellen. Anschliessend wird gezeigt, wie es gelingt, die Exponentialfunktion angenähert zu realisieren. Als Beifang finden wir dann die trigonometrischen Funktionen im Netz, das wir ausgeworfen haben. Etliche traditionelle Kenntnisse, die im Unterricht mit CAS-Rechnern zu verschwinden drohten, werden sich hier wieder als wesentlich in den Vordergrund schieben.

Beim Annähern der Quadratwurzeln genügt eine einzige Idee und die wiederholte schrittweise Verbesserung einer Anfangsschätzung. Beim angenäherten Berechnen der Logarithmen und der Exponentialfunktion kommen verschiedene Ideen aus der elementaren Analysis zum Zuge. Das Ziel der Überlegungen ist immer ein ganz konkreter Algorithmus, der auf einem programmierbaren Rechner eine Näherung schafft, deren Genauigkeit einen Vergleich mit jener einer Standardimplementation der entsprechenden Funktion auf dem Rechner wagen darf. Allerdings sind manche professionell entwickelte Algorithmen oft auf eine bestimmte Hardware massgeschneidert und in Bereichen optimiert, die hier nicht angesprochen werden können.

Voraussetzungen

Es wird angenommen, dass eine einfache Programmierumgebung zur Verfügung steht, die numerische Experimente erlaubt. Aus der Analysis werden Kenntnisse über Polynome und rationale Funktionen vorausgesetzt. Taylorentwicklung sollte intuitiv als 'verallgemeinerte Tangenten' eingeführt worden sein. Bei Bedarf können Faustregeln über die Abschätzung des Restgliedes bei der Taylorformel nachgeschoben werden. Eine gründliche Bearbeitung der Aufgaben durch die Lernenden selbst ist unabdingbar. Allerdings muss nicht jeder alles gemacht haben. Es ist sehr wohl denkbar, die Aufgaben an Gruppen zu delegieren, die diese aufbereiten und in Kurzvorträgen die Lösungen und die *Erfahrungen* im Plenum vorstellen. Dabei ist es wichtig zu beachten, dass gelegentlich mehrere analoge Fragen auftreten. Sie unterscheiden sich dann vielleicht in der Schwierigkeit etwas, aber sie erlauben eine Parallelbearbeitung und eine Diskussion der Ergebnisse vor dem Hintergrund der je eigenen Erfahrung. Ferner lassen sich die eigenen Ergebnisse mit denen anderer vergleichen oder auf Plausibilität prüfen. Für rege Diskussionen dürfte gesorgt sein.

2 Zum Beispiel Quadratwurzeln

Mit den arithmetischen Grundoperationen gelingt es gerade mal, lineare Gleichungssysteme zu lösen. Die einfachste nichtlineare Gleichung erfordert bereits eine Operation, die sich mit endlich vielen arithmetischen Operationen und mit rationalen Zahlen in der Regel nicht ausführen lässt. Es ist die reinquadratische Gleichung $x^2 = a > 0$.

Wenn es keine exakte Lösung gibt, so könnten Näherungswerte von Interesse sein. Ferner sind Verfahren gesucht, um Näherungen zu verbessern. Diese Idee hat eine lange Tradition. Sie geht mindestens bis in die babylonische Mathematik zurück, die sich bekanntlich erfolgreich mit dem Lösen quadratischer Gleichungen befasst hat. Über Beweise oder Begründungen ist aus babylonischer Zeit nichts bekannt. Erfolg rechtfertigt die Mittel. Die folgende heuristische Überlegung benutzt wesentlich Eigenschaften des arithmetischen und des geometrischen Mittels. Beide liessen sich durch Konstruktionen mit Zirkel und Lineal veranschaulichen. Das war in einer Zeit vor der Algebra sicher hilfreich.

Angenommen, wir schätzen die positive Lösung der Gleichung $x^2 = a$ mit einer Zahl $s > 0$. Dann gilt $s^2 \approx a$. Wenn s zu klein geschätzt wurde, muss der Schätzwert vergrössert werden. Nehmen wir einen zweiten Schätzwert t, der die lineare Gleichung $s \cdot t = a$ löst, so haben wir zwei Vorteile: Die Zahl $t = a/s$ lässt sich mit elementaren Mitteln einfach und exakt ausrechnen. Aber wichtiger: Das geometrische Mittel aus s und t ist die exakte Lösung $\sqrt{a}$. Allerdings können wir gerade diese Zahl ja nicht berechnen. Wenn aber s eine *gute* Schätzung ist, so erwarten wir, dass s und t nahe bei einander liegen. Das arithmetische Mittel $(s+t)/2$ liegt dann näher beim geometrischen Mittel, also bei der Lösung, als s oder t.

Dieses Vorgehen ist unter dem Namen *Heronverfahren* bekannt. Eine weitgehende Verallgemeinerung auf Gleichungen der Art $f(x) = 0$ mit einer stetig differenzierbaren Funktion f ist die *Newtoniteration.* Newtoniteration angewandt auf die Gleichung $x^2 - a = 0$ mit $a > 0$ entspricht dem Heronverfahren. Ein geeigneter Startwert s wird wiederholt durch die Verbesserung $s^* = s - f(s)/f'(s)$ ersetzt, bis ein vereinbartes Abbruchkriterium erreicht ist: entweder eine vorgegebene Maximalzahl von Versuchen oder das Unterschreiten einer vorgeschriebenen Genauigkeitsschranke für die Verbesserung $|f(s)/f'(s)|$.

Aufgaben

1. Realisieren Sie eine eigene Version `mysqrt()` einer numerischen Näherung für die Quadratwurzelfunktion.

2. Angenommen, $x^2 = 5$ soll ausgehend vom Startwert $s = 2$ mit dem Heronverfahren angenähert gelöst werden. Beobachten Sie das Muster der Ziffern in aufeinander folgenden Näherungen. Was ist bemerkenswert?

3. Warum gibt es keine Maschinenzahl, die $x^2 = 5$ exakt löst?

4. Welchen maximalen Approximationsfehler können Sie im Bereich $10^{-6} \leq x \leq 10^6$ für Ihre Version von `mysqrt()` garantieren? Wo ist der relative Fehler am grössten?

5. Wie lässt sich erreichen, dass die Quadratwurzel für beliebige positive Rechnerzahlen numerisch berechnet werden kann, wobei aber `mysqrt()` nur auf Zahlen im Bereich $[1/4, 4]$ auszuwerten ist?

6. Wie lässt sich der Approximationsfehler von `mysqrt()` abschätzen ohne Rückgriff auf eine Standardfunktion Ihrer Programmierumgebung?

3 Exponentialfunktion

Der natürliche Logarithmus ln und die Exponentialfunktion exp sind besonders wichtigste elementare Funktionen. Sie haben interessante Eigenschaften: Sie bilden ein Paar von zueinander inversen Funktionen. Der Logarithmus bildet die multiplikative Struktur der positiven reellen Zahlen auf die additive Struktur von $\mathbb{R}$ ab. Die Exponentialfunktion macht als Umkehrfunktion genau das Gegenteil.

Die Exponentialfunktion exp ist jene Lösung der Differentialgleichung $f' = f$, die $f(0) = 1$ erfüllt. Aus dieser Definition ergeben sich unmittelbar zwei Näherungen zum Berechnen von $\exp(x)$, wenn x nicht zu weit von 0 entfernt ist. Die erste Überlegung führt auf Potenzreihen und eine analytische Beschreibung von $\exp(x)$, die zweite entspricht einer numerischen Näherung an $\exp(x)$. Es wird je eine Methode skizziert, um die Exponentialfunktion exp durch Polynome auf einem begrenzten Intervall anzunähern.

3.1 Klassische Analysis

Aus der Differentialgleichung $f' = f$ und der Anfangsbedingung $f(0) := 1$ folgt rekursiv, dass für alle Ableitungen $f^{(n)}(0) = 1$ gilt. Daher gibt es zu exp die *Taylorpolynome* T_n mit

$$T_n(x) = \sum_{r=0}^{n} \frac{1}{r!}\, x^r$$

Die Koeffizienten $1/r!$ streben so schnell gegen 0, dass der Grenzwert $\lim_{n\to\infty} T_n(x)$ für jedes x existiert und die Funktion $\exp(x)$ definiert. Daher bieten sich Taylorpolynome als Option an, um zu gegebenem x den Funktionswert $\exp(x) \approx T_n(x)$ durch Auswerten des Taylorpolynoms angenähert zu berechnen.

3.2 Euler's einfache Numerik

Wer die *Differentialgleichung* $f' = f$ mit dem Anfangswert $f(0) = 1$ numerisch lösen möchte, kann dazu die ursprünglich von Euler betrachtete Methode verwenden. Nach x gelangt man von 0 aus der Reihe nach in n Schritten der Grösse x/n. Wegen $f'(0) = 1$ folgt $f(x/n) \approx f(0) + x/n \cdot f'(0) = 1 + x/n$. Wegen der Differentialgleichung ist nun auch $f'(x/n) \approx 1 + x/n$ und das Verfahren lässt sich von da aus über $n - 1$ weitere Schritte bis nach x fortsetzen. Beim Ausführen dieser Idee stösst man auf die Näherung $f(x) \approx (1 + x/n)^n$. Das lässt sich so lesen: Es gibt ein Polynom $e_n : x \mapsto (1 + x/n)^n$ vom Grade n, welches $\exp(x)$ angenähert berechnen kann. In der Tat lässt sich zeigen, dass für jedes gegebene x gilt

$$\lim_{n\to\infty} T_n(x) = \exp(x) = \lim_{n\to\infty} e_n(x)$$

Die beiden Ideen liessen sich kombinieren. Für alle Zahlen k gilt $\exp(x) = \exp(x/k)^k$ für alle x. Folglich könnte $(T_n(x/k))^k \approx \exp(x)$ eine effizientere Annäherung durch ein Polynom vom Grade $k \cdot n$ sein als $T_n(x)$ oder $e_n(x)$. Allerdings gelten diese Überlegungen bloss im Rahmen einer *fiktiven exakten Arithmetik mit reellen Zahlen, wie wir sie in der Analysis als gegeben annehmen.*

Aufgaben

Wir betrachten die Exponentialfunktion exp auf dem Intervall $-2 \le x \le 2$ und untersuchen, wie gut die Näherungspolynome T_n und e_n in der Lage sind, die in Ihrer Programmierumgebung vorhandene Realisierung von exp() anzunähern.

1. Untersuchen Sie die Graphen der Funktion

$$d_1 : x \mapsto e_n(x) - \exp(x) \;\; \text{für } n = 1, 2, 4, 8, 16, 32$$

   auf $[-2, 2]$. Wie gross sind die Extrema und wo treten sie auf?

2. Untersuchen Sie die Graphen der Funktion

$$d_2 : x \mapsto T_n(x) - \exp(x) \;\; \text{für } n = 1, 2, 4, 8, 16$$

   auf $[0, 2]$. Wie gross sind die Extrema und wo treten sie auf?

3. Untersuchen Sie die Graphen der Funktion $d_3 : x \mapsto (T_n(x/p))^p - \exp(x)$ mit $p = 256$ für $n = 1, 2, 4, 8$ auf $[0, 2]$. Wie gross sind die Extrema und wo treten sie auf? Sollte n eher vergrössert werden? Warum ist allgemein die Wahl $p = 2^r$, $r \in \mathbb{N}$ für den Parameter p günstig? Gibt es eine Begründung für $p = 256$? Lässt sich $p = 256$ ohne nennenswerte Einbusse an Genauigkeit auf $p = 128$ verringern?

4. Angenommen, die Funktionswerte $E_r = \exp(2^r)$ sind mit den verfügbaren Maschinenzahlen so genau als möglich bekannt für $r = 1, 2, 3, \ldots, 11$. [zB Mantisse mit 13(+3) Dezimalstellen, Exponent bis 10^{999}]

   (a) Formulieren Sie einen Algorithmus im Pseudocode, um $exp(k)$ für beliebige natürliche Zahlen $k \leq 2^{11}$ mit hoher Genauigkeit mit Hilfe der Liste $\{E_r\}$ anzunähern.

   (b) Erweitern Sie den Algorithmus so, dass $\exp x$ für beliebige reelle Zahlen x aus dem Bereich $0 \leq x \leq 2^{12}$ mit hoher Genauigkeit angenähert wird. Wie lässt sich $\exp(x)$ für $x < 0$ approximieren?

   (c) Welche absoluten und welche relativen Genauigkeiten können Sie im ganzen Definitionsbereich ihrer Näherungsfunktion garantieren?

Stellen Sie alle Ihre Folgerungen aus diesen Experimenten kurz und übersichtlich dar.

3.3 Rationale Approximation und Fixpunktiteration für Funktionen

Die Funktion $\exp : x \mapsto \exp(x)$ löst die Differentialgleichung $y' = y$ mit der Anfangsbedingung $y(0) = 1$. Daher gilt $\int_0^x \exp(t)\, dt = \exp(x) - 1$. Wenn nun das Integral in einem einzigen Eulerschritt angenähert wird, gilt $1 + \int_0^x \exp(t)\, dt \approx (1 + x)$ und entsprechend $1/\exp(x) = \exp(-x) = 1 + \int_0^{-x} \exp(t)\, dt \approx (1 - x)$, woraus sich

$$\frac{\exp(x)}{\exp(-x)} = \exp(2x) \approx \frac{1 + x}{1 - x}$$

ergibt. Damit lässt sich die erste Näherung $\exp(x) \approx 1 + x$ verbessern zu

$$\exp(x) \approx e_0(x) := \frac{1 + x/2}{1 - x/2}$$

Die Taylorentwicklung zeigt nun die vorerst bescheidene Güte dieser Approximation.

$$e_0(x) = \underline{1 + x + \frac{1}{2}x^2} + \frac{1}{4}x^3 + \frac{1}{8}x^4 \ldots$$

Aber der Quotient entspricht für $|x| < 1$ einer *nicht abbrechenden Potenzreihe, die für* $|x| < 1$ *konvergiert.* Die Näherung $e_0(x) \approx \exp(x)$ stimmt bis zum quadratischen Term überein. Das ist mehr als die erste Näherung aus dem Eulerschritt, aber doch noch zu bescheiden für unsere Zwecke. Insbesondere könnte die Nennernullstelle bei $x = 2$ Probleme verursachen. Um die Übereinstimmung zu steigern, benutzen wir den Ansatz

$$e_1 : x \mapsto \frac{1 + \frac{1}{2}x + a \cdot x^2}{1 - \frac{1}{2}x + a \cdot x^2}$$

mit einer noch zu bestimmenden Zahl a. Sie soll so bestimmt werden, dass die Taylorentwicklung von e_1 an der Stelle 0 mindestens bis zur dritten Ordnung mit jener von exp übereinstimmt.

Die Rechnung ergibt $a = 1/12$ und die Kontrolle mit Taylorentwicklung zeigt einen unerwarteten Erfolg:

$$e_1(x) = \frac{1 + \frac{1}{2}x + \frac{1}{12}x^2}{1 - \frac{1}{2}x + \frac{1}{12}x^2} \approx \underline{1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4} + \frac{1}{144}x^5 + \dots$$

Übereinstimmung mit der Taylorentwicklung von $\exp(x)$ bis und mit der Ordnung 4 und ein Fehler von $(1/144 - 1/120)x^5 = -x^5/720$ in der fünften Ordnung sind ersichtlich.

Da nahe bei $x = 0$ die Näherung $e_1(x) \approx \exp(x)$ gilt, erwarten wir mindestens für $x \approx 0$ die Beziehung $e_1(2x) \approx e_1(x)^2$ und zwar wiederum umso besser, je mehr die Funktionen $\exp : x \mapsto \exp(x)$ und $e_1 : x \mapsto e_1(x)$ einander gleichen.

Die Fixpunktgleichung $(f(x/2))^2 = f(x)$ wird von jeder Exponentialfunktion $\exp_b : x \mapsto b^x$ mit $b > 0$ erfüllt.

Fehlerabschätzungen und Experimente

Wir benutzen zum einfachen Experimentieren einen Algorithmus $\mathsf{e_1(x, n)}$ mit zwei Eingabeparametern: n bezeichnet die Anzahl der Bisektionen (und Verdoppelungen) und x ist der Wert der Funktionsvariablen von $e_1 : x \mapsto e_1(x)$.

```
e1(x, n)
    x/2^n → x
    1 + x/2 + x^2/12 → h(x)
    h(x)/h(−x) → q
    For j = 1 to n
        q ∗ q → q
    EndFor
    return q
End
```

Eine Implementation auf einem programmierbaren Rechner hat folgende Ergebnisse gebracht:

n	1	8	12	16	20
$e1(1, n)$	2.71419	2.718280905	2.718281646	2.71828181771	2.7182818049
Fehler	$-3.996 \cdot 10^{-3}$	$-9.23 \cdot 10^{-7}$	$-1.82 \cdot 10^{-7}$	$-5.76 \cdot 10^{-8}$	$-2.36 \cdot 10^{-8}$

Die Tabelle zeigt, dass sich mit dem verwendeten Verfahren die Genauigkeit auch mit $n = 20$ Halbierungsschritten *nicht beliebig steigern* lässt. Bei der Berechnung der Hilfsgrössen $h(x)$ stehen Diskretisierungsfehler und Stellenauslöschung oder Rundungsfehler im Widerstreit. Diese numerische Schwachstelle lässt sich vermeiden.

Eine bessere Fixpunktiteration

Aus $(e_1(x/2))^2 = e_1(x)$ gewinnt man:

$$2 \cdot k(x/2) + (k(x/2))^2 = k(x).$$

Folgende Umformung verbessert die Ergebnisse

$$e_1(x) = \frac{1 + \frac{1}{2}x + \frac{1}{12}x^2}{1 - \frac{1}{2}x + \frac{1}{12}x^2} = 1 + \frac{x}{1 - \frac{1}{2}x + \frac{1}{12}x^2} = 1 + k(x)$$

Die Fixpunktgleichung für e_1 lässt sich umwandeln in eine Fixpunktgleichung für k:
Aus $(e_1(x/2))^2 = e_1(x)$ gewinnt man $2 \cdot (1 + k(x/2)) * k(x/2) = k(x)$. Damit lässt sich der Algorithmus verbessern:

```
ee1(x, n)
   x/2^n → x
   x/(1 − x/2 + x^2/12) → k
   For j = 1 to n
      (2 + k) ∗ k → k
   next j
   return 1 + k
End
```

Versuche zeigen nun, dass die Massnahme erfolgreich ist, denn $ee1(x, 10)$ liefert für alle x im Intervall $[-1, 1]$ eine Übereinstimmung mit der in einem programmierbaren Rechner implementierten Exponentialfunktion. Die beobachteten Beträge der Abweichungen sind durchwegs geringer als 10^{-13}.

3.4 Anwendung: Trigonometrische Funktionen

Aus dem Vergleich der Taylorreihen für exp, cos, sin ergibt sich die *Eulerrelation*

$$\exp(i \cdot t) = \cos(t) + i \cdot \sin(t) \ ,$$

wenn man für die Variable in der Exponentialfunktion die reinimaginäre Zahl $i \cdot t$ einsetzt und im Ergebnis Real- und Imaginärteil trennt. Jedes Verfahren, das die Exponentialfunktion exp in einem reellen Intervall numerisch approximiert, kann versuchsweise auch auf komplexe Eingaben angewendet werden.

Dieses Vorgehen setzt voraus, dass die Programmierumgebung die komplexe Arithmetik und den Datentyp der komplexen Rechnerzahlen kennt. Alle Näherungsverfahren für $\exp(x)$ mit rationalen Funktionen (also speziell auch mit Polynomen) eignen sich für einen solchen Versuch.

Ein Testfall: $\exp(i\pi/4)$ Der Algorithmus *ee1*(x, n) ist auch auf komplexe Eingaben x anwendbar. Vorsicht ist geboten, weil in $\mathbb{C}$ Nennernullstellen von $k(x)$ bei $3 \pm i\sqrt{3}$ auftreten. Versuche mit $ee1(i \cdot x, 10)$ und x aus dem Intervall $[-\pi/4, \pi/4]$ zeigen wiederum eine Genauigkeit besser als 10^{-13} im Vergleich zu den implementierten Standardfunktionen sin und cos im betrachteten Intervall.

Hinweis: In [www.swisseduc.ch/Mathematik] zeigt der Beitrag *Cosinus und Sinus numerisch effizient annähern* eine ganz andere numerischen Approximation der Grundfunktionen Cosinus und Sinus in der komplexen Ebene $\mathbb{C}$.

Die Grundlage ist dort eine Bewegung auf dem Einheitskreis mit Einheitsgeschwindigkeit. Dazu wird eine einfache Differentialgleichung in der komplexen Ebene $\mathbb{C}$ formuliert und auf einsichtige Art *numerisch behandelt.*

Einige Beispiele zu Aufgaben

1. Angenommen, die Funktion η ist eine numerische Näherung an die Exponentialfunktion, die auch $\exp(i \cdot t)$ annähert für $0 \leq t \leq \pi/4$. Warum ist dies ausreichend, um Näherungen für $\cos(t)$ und $\sin(t)$ für $-\pi \leq t \leq \pi$ und damit im Prinzip für 'alle' t herzuleiten. Notieren Sie eine Näherung $c : t \mapsto c(t)$ in Pseudocode für die Cosinusfunktion und eine analoge Näherung für die Sinusfunktion ausgehend von der hypothetischen Funktion η.

2. Realisieren Sie Näherungen von cos() und sin() aufgrund der bereits vorhandenen Näherungen für exp

   (a) mit der Taylornäherung $T_8(x)$ für exp

   (b) mit der Taylornäherung und der Verbesserung $ee : x \mapsto (T_8(x/256))^{256}$

   (c) $ee1 : x \mapsto ee1(x)$

3. Wie lässt sich eine Approximation der Exponentialfunktion in $\mathbb{C}$ testen?

   (a) mit der in der Programmierumgebung vorhandenen Standardfunktion.

   (b) ohne Rückgriff auf die Standardfunktionen der Programmierumgebung.