Projekt I/5: Approximation von Standardfunktionen

Dieses Projekt hat zum Ziel, einige der so genannten mathematischen Standardfunktionen wie $\exp(x)$, $\ln(x)$, $\sin(x)$, ... numerisch zu approximieren.

Diese Standardfunktionen sind in den meisten Implementierungen der gängigen Programmiersprachen fix und fertig bereitgestellt, manche sogar bereits auf Hardwareebene (z.B. als Maschinenintruktion für $\sqrt{x}$). In den folgenden Beispielen sollen davon unabhängige numerische Approximationen gebastelt werden. Diese Aufgabenstellung stellt ein schönes Übungsgelände für verschiedene numerische Techniken dar, wie z.B. Summation, Argumentreduktion, partielle doppelt Genauigkeit etc., und man muss mit den dabei auftretenden Rechenfehlereffekten zurechtkommen. Außerdem ist es denkbar, dass man auf exotischeren Rechnerarchitekturen tatsächlich gezwungen ist, eigene Routinen für die Standardfunktionen zu entwickeln.

Manche der relevanten numerischen Techniken, wie Reihenentwicklung, sind bei jedem der einzelnen Beispiele von Bedeutung; manches andere ist Beispiel-spezifisch. Für die Approximation werden jeweils Polynome verwendet, die aus abgebrochenen Taylorreihen gewonnen werden. Es gibt auch polynomiale Approximationen (z.B. Chebyshev-Interpolierende), die für diesen Zweck besser geeignet sind; diese werden jedoch hier nicht einbezogen.

Zu beachten:

- Grafische Visualisierung der Ergebnisse ist sinnvoll und erwünscht.
- Der subnormale Gleitpunktzahlenbereich soll bei diesem Projekt grundsätzlich außer Betracht bleiben.
- Mit $\mathbb{F}$ bezeichnen wir den zugrundeliegenden Bereich von Gleitpunktzahlen. Als Rechnerarithmetik wird grundsätzlich doppelt genaue IEEE-Arithmetik (d.h. etwa `double` in MATLAB auf einem PC, d.h. ca. 15 signifikante Dezimalstellen) vorausgesetzt, dies ist aber nicht wirklich wesentlich.
- Mit *eps* bezeichnen wir die relative Maschinengenauigkeit für diese Arithmetik. Die in den Beispielen zu konstruierenden Approximationen sollen möglichst bis auf einen relativen Fehler der Größenordnung 100 *eps* genau sein; aus Gründen des Rechenaufwandes kann aber auch eine moderatere Fehlertoleranz K *eps* gewählt werden, mit $K > 100$.

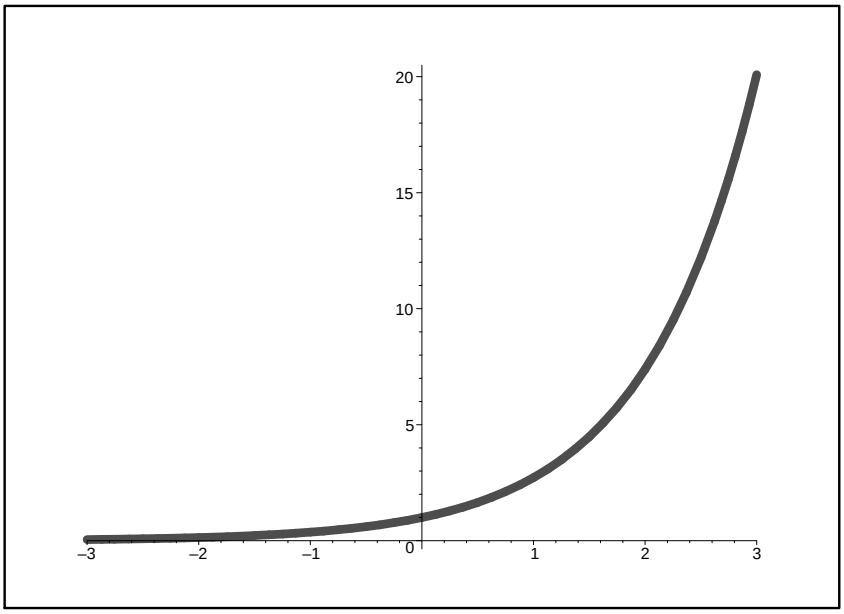

Abbildung 1: Exponentialfunktion e^x

Beispiel 1: Exponentialfunktion

Es soll eine Funktionsprozedur für die Exponentialfunktion $\exp(x) = e^x$ entwickelt werden. Die Funktion wird dabei durch die ersten $n+1$ Summanden ihrer Taylorreihe bezüglich der Stelle $x_0 = 0$ ersetzt:

$$e^x \approx \sum_{i=0}^{n} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots + \frac{x^n}{n!}$$

(1.1) Man formuliere die Aufgabenstellung als numerisches Problem für einen konkreten Rechner:

- Genauigkeitsforderung: Es wird eine relative Genauigkeit von mindestens 100 *eps* angestrebt.
- Für welchen Bereich von x-Werten ist es (bei gegebener Arithmetik) überhaupt sinnvoll, eine Approximaton zu berechnen ('numerischer Definitionsbereich')? Welche Ausnahmesituation(en) liegt(liegen) außerhalb dieses Bereiches vor? Gibt es Symmetrien, so dass dieser Bereich weiter reduziert werden kann?
- Für welche speziellen Argumente $x \in \mathbb{F}$ ist der Funktionswert trivial?

(1.2) Wie viele Glieder der Taylorreihe muss man mindestens summieren, damit man die gewünschte Genauigkeit erreicht? Man führe eine Abschätzung für beliebige Argumente x aus dem numerischen Definitionsbereich durch und stelle diese Information in geeigneter Weise grafisch dar.

Dabei wird exakte Rechnung vorausgesetzt: Die Rechenfehler, die beim Aufsummieren der Reihe entstehen, werden vernachlässigt. Der Reihenrest,

$$e^x - \sum_{i=0}^{n} \frac{x^i}{i!} = \frac{x^{n+1}}{(n+1)!}\, e^{\theta x} \qquad \text{mit} \quad \theta \in [0,1],$$

entspricht dem absoluten Verfahrensfehler.

(1.3) Für einige konkrete Argumente $x \in \mathbb{F}$,

$$x \;=\; 1,\; -1,\; 10,\; -10,\; 20,\; -20,\; 100,\; -100$$

führe man die Summation dieser endlichen Reihen in natürlicher Reihenfolge durch und ermittle die relativen Fehler der so erhaltenen Resultate. (Als Bezugsgröße verwendet man den von der Standardprozedur gelieferten Wert.)

Wird eine relative Genauigkeit erreicht, die dem unter (1.2) angegebenen Verfahrensfehler entspricht?

(1.4) Man gebe eine Begründung für die beobachteten Phänomene und vergleiche die Qualität der obigen Resultate mit jener, die man bei Summation in umgekehrter Reihenfolge erhält. Hat man durch diese Maßnahme die Fehler bei den betragsgroßen negativen Argumenten in den Griff bekommen?

(1.5) Man entwickle folgende praktikable Variante, die auf dem Prinzip der *Argumentreduktion* beruht: Man approximiert die Funktion nur auf einem kleinen Standardintervall, nämlich $[1/e, 1]$, und verwendet spezielle Eigenschaften der Exponentialfunktion ($e^{x+y} = e^x \cdot e^y$), um andere Argumente auf diesen Fall zurückzuführen.

Im folgenden bezeichne $\mathbf{ln2} \in \mathbb{F}$ die numerische Approximation von $\ln 2$ in der gegebenen Arithmetik.

a) Ist $x \in [0, \mathbf{ln2}]$, so wird e^x mit dem obigen Algorithmus ermittelt, wobei die Taylorreihe in der umgekehrten Reihenfolge summiert wird. (Es empfiehlt sich ein zusätzlicher Summand als 'Genauigkeitsreserve'.) Speziell ist $e^0 = 1$.

b) Ist $x > \mathbf{ln2}$, dann stellt man x dar als

$$x \;=\; k\,\mathbf{ln2} + r, \qquad \text{mit} \quad k \in \mathbb{N}, \quad 0 \le r < \mathbf{ln2},$$

und verwendet die Identität

$$e^x \;=\; e^{k\,\ln 2 + r} = 2^k\, e^r.$$

Man berechnet also e^r mit der Prozedur aus a) und führt anschließend die Multiplikation mit 2^k durch. (Liegt eine Arithmetik mit der Basis $b = 2$ vor, so erfolgt diese Multiplikation sogar rechenfehlerfrei. Für eine andere Basis b genügt es, in a) das rechte Intervallende **ln2** durch $\ln b$ zu ersetzen, um den gleichen Effekt zu erzielen.)

c) Ist $x < 0$, so berechnet man zuerst $e^{|x|}$ nach a) oder b) und bildet anschließend $e^x = 1/e^{|x|}$.

(1.6) Mit dieser Variante berechne man neuerlich die Werte der Exponentialfunktion für $x = 1,\; -1,\; 10,\; -10,\; 20,\; -20,\; 100,\; -100$. Hat sich die Qualität der Werte für große negative Exponenten verbessert?

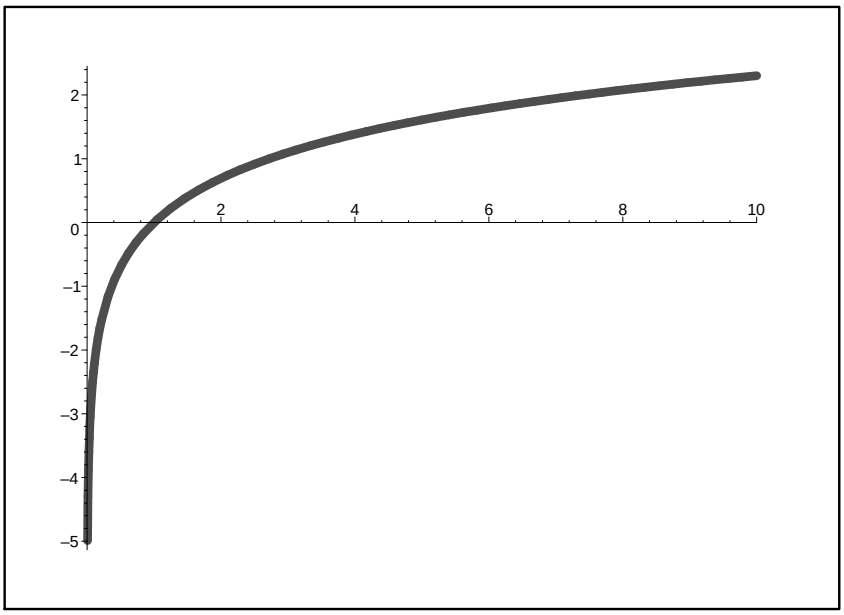

Abbildung 2: Natürlicher Logarithmus $\ln x$

Beispiel 2: Natürlicher Logarithmus

Es soll eine Funktionsprozedur für den natürlichen Logarithmus $\ln x$ (Inverse der Exponentialfunktion, definiert für $x > 0$) entwickelt werden. Die Funktion soll dabei durch die ersten n Summanden ihrer Taylorreihe bezüglich der Stelle $x_0 = 1$ ersetzt werden: Für $x = 1 + \delta$ ist

$$\ln(1+\delta) \approx \sum_{i=1}^{n} (-1)^{i-1} \frac{\delta^i}{i} = \delta - \frac{\delta^2}{2} + \frac{\delta^3}{3} - \frac{\delta^4}{4} + \ldots + (-1)^{n-1} \frac{\delta^n}{n},$$

diese Reihe konvergiert jedoch nur für $0 < x \leq 2$, d.h. für $|\delta| \leq 1$, $\delta \neq -1$ (siehe unten).

(2.1) Man formuliere die Aufgabenstellung als numerisches Problem für einen konkreten Rechner:

- Genauigkeitsforderung: Es wird eine relative Genauigkeit von mindestens 100 *eps* angestrebt.
- Für welchen Bereich von x-Werten ist es (bei gegebener Arithmetik) überhaupt sinnvoll, eine Approximaton zu berechnen ('numerischer Definitionsbereich')? Welche Ausnahmesituation(en) liegt(liegen) außerhalb dieses Bereiches vor? Gibt es Symmetrien, so dass dieser Bereich weiter reduziert werden kann?
- Für welche speziellen Argumente $x \in \mathbb{F}$ ist der Funktionswert trivial?

(2.2) Wir betrachten zunächst Argumente $0 < x \leq 1$, d.h. $\delta \in (-1, 0]$, für die die obige Taylorreihe konvergiert. Wie viele Glieder der Taylorreihe muss man mindestens summieren, damit man die gewünschte Genauigkeit erreicht? Man führe eine Abschätzung für beliebige Argumente $x \in (0, 1]$ durch und stelle diese Information in geeigneter Weise grafisch dar. Wo ist der 'problematische Bereich' von x-Werten?

Dabei wird exakte Rechnung vorausgesetzt: Die Rechenfehler, die bei dem Aufsummieren der Reihe entstehen, werden vernachlässigt. Der Reihenrest,

$$\ln(1+\delta) - \sum_{i=1}^{n} (-1)^{i-1} \frac{\delta^i}{i} \;=\; (-1)^n \frac{\delta^{n+1}}{n+1} \cdot \frac{1}{(1+\theta\delta)^{n+1}} \qquad \text{mit} \quad \theta \in [0,1],$$

entspricht dem absoluten Verfahrensfehler.

(2.3) Für einige konkrete Argumente $x \in \mathbb{F}$,

$$x \;=\; 2^{-k}, \quad k = 1, 2, \ldots$$

führe man die Summation dieser endlichen Reihen in natürlicher Reihenfolge durch und ermittle die relativen Fehler der so erhaltenen Resultate. (Als Bezugsgröße verwendet man den von der Standardprozedur gelieferten Wert.)

Wird eine relative Genauigkeit erreicht, die dem in (2.2) angegebenen Verfahrensfehler entspricht?

(2.4) Man gebe eine Begründung für die beobachteten Phänomene und vergleiche die Qualität der obigen Resultate mit jener, die man bei Summation in umgekehrter Reihenfolge erhält. Ist ein Unterschied erkennbar?

(2.5) Man entwickle folgende praktikable Variante, die auf dem Prinzip der *Argumentreduktion* beruht: Man approximiert die Funktion nur auf einem kleinen Standardintervall und verwendet spezielle Eigenschaften der Logarithmusfunktion um andere Argumente auf diesen Fall zurückzuführen.

Im folgenden bezeichne $\mathbf{e} \in \mathbb{F}$ die numerische Approximation von $e = \exp(1)$ in der gegebenen Arithmetik.

a) Ist $x \in [1/\mathbf{e}, 1]$, so wird $\ln x$ mit dem obigen Algorithmus ermittelt, wobei die Taylorreihe in der umgekehrten Reihenfolge summiert wird. (Es empfiehlt sich ein zusätzlicher Summand als 'Genauigkeitsreserve'.) Speziell ist $\ln 1 = 0$.

(Eine mögliche Alternative besteht darin, $[1, \mathbf{e}]$ als Basisintervall zu wählen, was im weiteren entsprechende Modifikationen nach sich zieht.)

b) Ist $0 < x < 1/\mathbf{e}$, dann multipliziert man x so lange[1] mit dem Faktor $\mathbf{e}$, bis (nach k-facher Multiplikation mit $\mathbf{e}$) gilt

$$r \;:=\; \mathbf{e}^k \cdot x \in [1/\mathbf{e}, 1],$$

und verwendet die Identität

$$\ln x \;=\; \ln(r \cdot e^{-k}) \;=\; \ln r - k.$$

Man berechnet also $\ln r$ mit der Prozedur aus a) (mit $\delta = r - 1$) und subtrahiert anschließend k.

c) Ist $x > 1$, so berechnet man zuerst $\ln(1/x)$ nach a) oder b) und bildet anschließend $\ln x = -\ln(1/x)$.

[1] Das geht auch etwas eleganter und genauer. Man überlege sich eine Variante – dabei darf aber natürlich keine Auswertung eines Logarithmus vorkommen!

Mit dieser Variante berechne man neuerlich die Werte der Logarithmusfunktion für Argumente $x = 2^{-k}$ (vgl. (2.3)) und bestimme den Fehler dieser Approximationen. Was ist zu beobachten? (... Genauigkeit, Rechenaufwand)

(2.6) a) Für Argumente $x < 1/e$ nahe an $1/e$ tritt bei bei der Berechnung des zum reduzierten Argument $r = \mathbf{e}\,x$ gehörigen Wertes $\delta = r - 1$ *Auslöschung* auf, weil r mit einem Rundungsfehler behaftet ist und $r \approx 1$ ist.

Man diskutiere die Auswirkung dieses Auslöschungseffektes auf die Genauigkeit der resultierenden Approximation für $\ln x$, allenfalls unterstützt durch Tests. Besteht hier ein Handlungsbedarf?

b) Für Argumente $x > 1$ nahe an 1 erfordert die Argumentreduktion gemäß (2.5) die Berechnung von $1/x < 1$ (nahe an 1), und der berechnete Wert von $1/x$ ist natürlich mit einem Rundungsfehler behaftet.

Man analysiere diese Situation und treffe die entsprechenden Maßnahmen hinsichtlich der Implementierung, mit begleitenden numerischen Experimenten.

(2.7) Man bestimme die relative Konditionszahl der Auswertung von $\ln x$ für Argumente $x < 1$ nahe an 1. Diese Konditionszahl ist groß (check!), aber für Maschinenzahlen $x \in \mathbb{F}$ (kein Datenfehler) erwartet man von einer guten Implementierung von ln, dass sie auch hier einen genauen Wert liefert.

Ist die Auswertung von $\ln x$ für $1 \approx x \in \mathbb{F}$ mittels der obigen Taylorreihe in diesem Sinne numerisch stabil? (Man betrachte insbesondere den Extremfall $x = 1 - O(eps)$, teste und analysiere.)

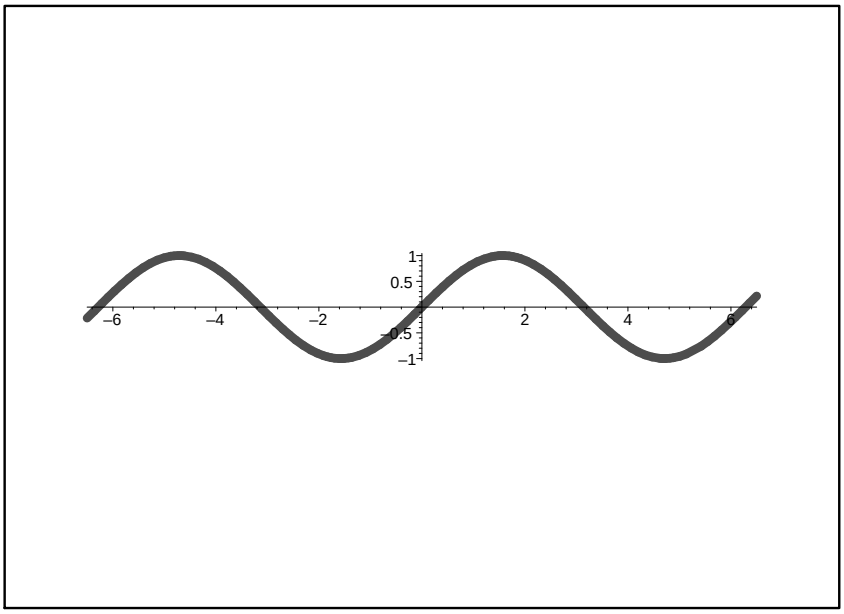

Abbildung 3: Sinusfunktion $\sin x$

Beispiel 3: Sinusfunktion

Es soll eine Funktionsprozedur für die Sinusfunktion $\sin x$ entwickelt werden. Die Funktion wird dabei durch die ersten $n+1$ Summanden ihrer Taylorreihe bezüglich der Stelle $x_0 = 0$ ersetzt:

$$\sin x \;\approx\; \sum_{i=0}^{n} (-1)^i \frac{x^{2i+1}}{(2i+1)!} \;=\; x - \frac{x^3}{6} + \frac{x^5}{120} + \ldots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

(3.1) Man formuliere die Aufgabenstellung als numerisches Problem für einen konkreten Rechner:

- Genauigkeitsforderung: Es wird eine relative Genauigkeit von mindestens $100\,eps$ angestrebt.
- Für welchen Bereich von x-Werten ist es (bei gegebener Arithmetik) überhaupt sinnvoll, eine Approximaton zu berechnen ('numerischer Definitionsbereich')? Welche Ausnahmesituation(en) liegt(liegen) außerhalb dieses Bereiches vor? Gibt es Symmetrien, so dass dieser Bereich weiter reduziert werden kann?
- Für welche speziellen Argumente $x \in \mathbb{F}$ ist der Funktionswert trivial?

(3.2) Wie viele Glieder der Taylorreihe muss man mindestens summieren, damit man die gewünschte Genauigkeit erreicht? Man führe eine Abschätzung für beliebige Argumente x aus dem numerischen Definitionsbereich durch und stelle diese Information in geeigneter Weise grafisch dar.

Dabei wird exakte Rechnung vorausgesetzt: Die Rechenfehler, die bei dem Aufsummieren der Reihe entstehen, werden vernachlässigt. Der Reihenrest,

$$\sin x - \sum_{i=0}^{n} (-1)^i \frac{x^{2i+1}}{(2i+1)!} \;=\; (-1)^{n+1} \frac{x^{2n+3}}{(2n+3)!} \cos(\theta x), \qquad \text{mit} \quad \theta \in [0,1],$$

entspricht dem absoluten Verfahrensfehler.

(3.3) Für einige konkrete Argumente $x \in \mathbb{F}$,

$$x = 2^k, \quad k = 0, 1, 2, \ldots, k_{max},$$
$$x = 710,$$

führe man die Summation dieser endlichen Reihen in natürlicher Reihenfolge durch und ermittle die relativen Fehler der so erhaltenen Resultate. (Als Bezugsgröße verwendet man den von der Standardprozedur gelieferten Wert.) Dabei wähle man k_{max} 'vernünftig' (vgl. (3.2)!).

Wird eine relative Genauigkeit erreicht, die dem in (3.2) angegebenen Verfahrensfehler entspricht? Welche Argumentwerte sind besonders 'kritisch'?

(3.4) Man gebe eine Begründung für die beobachteten Phänomene und vergleiche die Qualität der obigen Resultate mit jener, die man bei Summation in umgekehrter Reihenfolge erhält. Hat man durch diese Maßnahme die Fehler bei den betragsgroßen Argumenten in den Griff bekommen?

(3.5) Man entwickle folgende praktikable Variante, die auf dem Prinzip der *Argumentreduktion* beruht: Man approximiert die Funktion nur auf dem Intervall $[0, 2\pi]$ und verwendet spezielle Eigenschaften der Sinusfunktion, insbesondere deren Periodizität, um andere Argumente auf diesen Fall zurückzuführen.

Im folgenden bezeichne $\mathbf{pi} \in \mathbb{F}$ die numerische Approximation von π in der gegebenen Arithmetik.

a) Ist $x \in [0, \mathbf{pi}]$, so wird $\sin x$ mit dem obigen Algorithmus ermittelt, wobei die Taylorreihe in der umgekehrten Reihenfolge summiert wird. (Es empfiehlt sich ein zusätzlicher Summand als 'Genauigkeitsreserve'.)

b) Der Fall $x \in (\mathbf{pi}, 2\mathbf{pi}]$, wird unmittelbar auf Fall a) zurückgeführt (wie?).

c) Ist $x > 2\,\mathbf{pi}$, dann stellt man x dar als

$$x = 2\,k\,\mathbf{pi} + r, \qquad \text{mit} \quad k \in \mathbb{N}, \quad 0 \le r < 2\,\mathbf{pi},$$

und verwendet die Periodizität von sin:

$$\sin(2k\pi + r) = \sin r.$$

Man bestimmt also das betreffende k, berechnet $r := x - 2\,k\,\mathbf{pi}$ und anschließend $\sin r$ gemäß $a), b)$.

d) Für $x < 0$ verwendet man die Schiefsymmetrie der Sinusfunktion: $\sin x = -\sin(-x)$.

(3.6) Mit der Variante (3.5) berechne man neuerlich die Werte der Sinusfunktion für die gleichen Argumente wie unter (3.3). Was hat sich verbessert, bzw. wo treten noch Ungenauigkeiten auf? Man achte auf die erreichte relative Genauigkeit!

Das spezielle Argument $x = 710$ steht stellvertretend für betragsgroße Argumente, bei denen $\sin x$ sehr betragsklein ist. (Beachte: $710 \approx 2\pi \cdot 113.000009\ldots$!) Diese Konstellation soll jetzt noch genauer untersucht werden.

– Wodurch ist bei solchen Argumenten die numerische Ungenauigkeit des unter (3.5) konstruierten Algorithmus verursacht? (Je kleiner $|\sin x|$ und je größer k, desto ausgeprägter ist diese Ungenauigkeit.)

- Kann man diese Ungenauigkeit dadurch beseitigen, dass man $\sin r$ 'exakt' auswertet (also mit der Standardprozedur für sin anstatt mit der selbstgebastelten Operation)?
- Welche Operation müsste man mit *höherer Genauigkeit* ausführen, um diese numerische Ungenauigkeit zu mildern bzw. zu beseitigen? (Begründung!)
- Für $\sin 710$ soll nun ein genauerer Wert in folgender Weise ermittelt werden: Man berechnet das reduzierte Argument r in erhöhter Genauigkeit (etwa mit 25 Dezimalstellen, mit Hilfe von MAPLE) übernimmt den sich daraus durch Rundung ergebenden Wert in das eigene Programm, und wertet dann sin aus.

  Man berechne die beiden Approximationen[2] für $\sin 710$, die sich mit diesen beiden Versionen von r ergeben ('normal' bzw. mit erhöhter Genauigkeit bestimmt), vergleiche diese mit dem 'exakten' Wert für $\sin 710$ und interpretiere die Resultate.
- Man bestimme auch die relative Konditionszahl der Auswertung von $\sin x$ für das 'kritische' Argument $x = 710$. Diese Konditionszahl ist groß (check!), aber da ja x exakt als Maschinenzahl gegeben ist (kein Datenfehler), erwartet man von einer guten Implementierung von sin, dass sie auch hier einen genauen Wert liefert.

(3.7*) Fleißaufgabe (nichttrivial!)

Unter (3.6) wurde das reduzierte Argument r für den konkreten Wert $x = 710$ mittels erhöhter Genauigkeit berechnet. Man versuche eine Methode zu finden und in MATLAB zu codieren, die für 'beliebige' (d.h. nicht allzu große) Argumente x (etwa $|x| \leq 10^5$) das zugehörige r möglichst genau berechnet (bis auf einen unvermeidlichen relativen Fehler der Größenordnung *eps*).

[2] Um den Einfluss der verschiedenen Fehlerquellen sauber zu trennen, soll bei diesem Vergleich der Wert von $\sin r$ der Einfachheit halber mit der 'exakten' Standardfunktion sin ausgewertet werden, und nicht mit der selbstgebastelten Approximation.

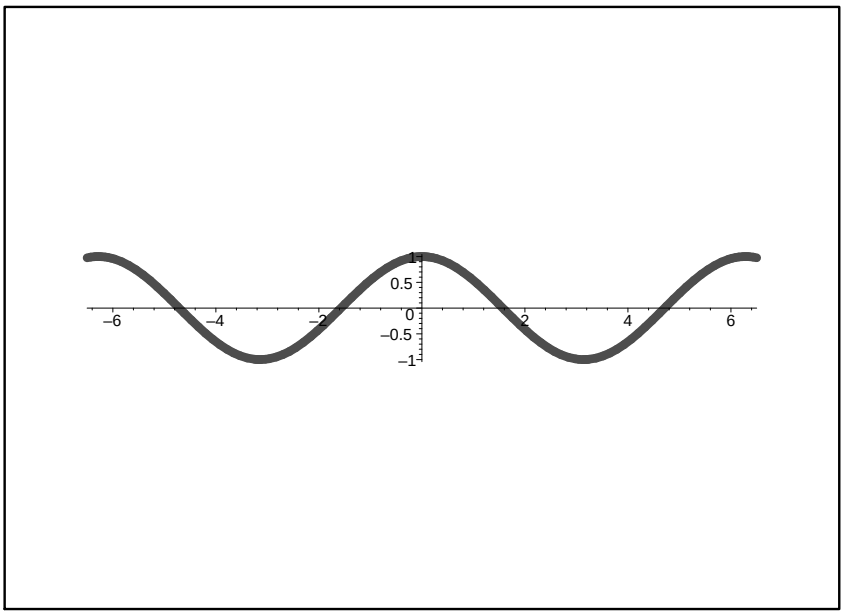

Abbildung 4: Cosinusfunktion $\cos x$

Beispiel 4: Cosinusfunktion

Es soll eine Funktionsprozedur für die Cosinusfunktion $\cos x$ entwickelt werden. Die Funktion wird dabei durch die ersten $n+1$ Summanden ihrer Taylorreihe bezüglich der Stelle $x_0 = 0$ ersetzt:

$$\cos x \approx \sum_{i=0}^{n} (-1)^i \frac{x^{2i}}{(2i)!} = 1 - \frac{x^2}{2} + \frac{x^4}{24} + \ldots + (-1)^n \frac{x^{2n}}{(2n)!}$$

(4.1) Man formuliere die Aufgabenstellung als numerisches Problem für einen konkreten Rechner:

- Genauigkeitsforderung: Es wird eine relative Genauigkeit von mindestens 100 *eps* angestrebt.
- Für welchen Bereich von x-Werten ist es (bei gegebener Arithmetik) überhaupt sinnvoll, eine Approximaton zu berechnen ('numerischer Definitionsbereich')? Welche Ausnahmesituation(en) liegt(liegen) außerhalb dieses Bereiches vor? Gibt es Symmetrien, so dass dieser Bereich weiter reduziert werden kann?
- Für welche speziellen Argumente $x \in \mathbb{F}$ ist der Funktionswert trivial?

(4.2) Wie viele Glieder der Taylorreihe muss man mindestens summieren, damit man die gewünschte Genauigkeit erreicht? Man führe eine Abschätzung für beliebige Argumente x aus dem numerischen Definitionsbereich durch und stelle diese Information in geeigneter Weise grafisch dar.

Dabei wird exakte Rechnung vorausgesetzt: Die Rechenfehler, die bei dem Aufsummieren der Reihe entstehen, werden vernachlässigt. Der Reihenrest,

$$\cos x - \sum_{i=0}^{n} (-1)^i \frac{x^{2i}}{(2i)!} = (-1)^{n+1} \frac{x^{2n+2}}{(2n+2)!} \cos(\theta x), \qquad \text{mit} \quad \theta \in [0,1],$$

entspricht dem absoluten Verfahrensfehler.

(4.3) Für einige konkrete Argumente $x \in \mathbb{F}$,

$$a) \quad x = 2^k, \quad k = 0, 1, 2, \ldots, k_{max},$$
$$b) \quad x = 887.5,$$

führe man die Summation dieser endlichen Reihen in natürlicher Reihenfolge durch und ermittle die relativen Fehler der so erhaltenen Resultate. (Als Bezugsgröße verwendet man den von der Standardprozedur gelieferten Wert.) Dabei wähle man k_{max} 'vernünftig' (vgl. (4.2)!).

Wird eine relative Genauigkeit erreicht, die dem in (4.2) angegebenen Verfahrensfehler entspricht? Welche Argumentwerte sind besonders 'kritisch'? Was passiert bei $x = 887.5$?

(4.4) Man gebe eine Begründung für die beobachteten Phänomene und vergleiche die Qualität der obigen Resultate mit jener, die man bei Summation in umgekehrter Reihenfolge erhält. Hat man durch diese Maßnahme die Fehler bei den betragsgroßen Argumenten in den Griff bekommen?

(4.5) Man entwickle folgende praktikable Variante, die auf dem Prinzip der *Argumentreduktion* beruht: Man approximiert die Funktion nur auf dem Intervall $[0, 2\pi]$ und verwendet spezielle Eigenschaften der Cosinusfunktion, insbesondere deren Periodizität, um andere Argumente auf diesen Fall zurückzuführen.

Im folgenden bezeichne $\mathbf{pi} \in \mathbb{F}$ die numerische Approximation von π in der gegebenen Arithmetik.

a) Ist $x \in [0, \mathbf{pi}]$, so wird $\cos x$ mit dem obigen Algorithmus ermittelt, wobei die Taylorreihe in der umgekehrten Reihenfolge summiert wird. (Es empfiehlt sich ein zusätzlicher Summand als 'Genauigkeitsreserve'.)

b) Der Fall $x \in (\mathbf{pi}, 2\mathbf{pi}]$, wird unmittelbar auf Fall a) zurückgeführt (wie?).

c) Ist $x > 2\,\mathbf{pi}$, dann stellt man x dar als

$$x = 2\,k\,\mathbf{pi} + r, \qquad \text{mit} \quad k \in \mathbb{N}, \quad 0 \le r < 2\,\mathbf{pi},$$

und verwendet die Periodizität von cos:

$$\cos(2k\pi + r) = \cos r.$$

Man bestimmt also das betreffende k, berechnet $r := x - 2\,k\,\mathbf{pi}$ und anschließend $\sin r$ gemäß $a), b)$.

d) Für $x < 0$ verwendet man die Symmetrie der Cosinusfunktion: $\cos x = \cos(-x)$.

(4.6) Mit der Variante (4.5) berechne man neuerlich die Werte der Cosinusfunktion für die gleichen Argumente wie unter (4.3). Was hat sich verbessert, bzw. wo treten noch Ungenauigkeiten auf? Man achte auf die erreichte relative Genauigkeit!

Das spezielle Argument $x = 887.5$ steht stellvertretend für betragsgroße Argumente, bei denen $\cos x$ sehr betragsklein ist. (Beachte: $887.5 - \pi/2 \approx 2\pi \cdot 141.00001\ldots$!) Diese Konstellation soll jetzt noch genauer untersucht werden.

– Wodurch ist bei solchen Argumenten die numerische Ungenauigkeit des unter (4.5) konstruierten Algorithmus verursacht? (Je kleiner $|\cos x|$ und je größer k, desto ausgeprägter ist diese Ungenauigkeit.)

- Kann man diese Ungenauigkeit dadurch beseitigen, dass man $\cos r$ 'exakt' auswertet (also mit der Standardprozedur für cos anstatt mit der selbstgebastelten Operation)?
- Welche Operation müsste man mit *höherer Genauigkeit* ausführen, um diese numerische Ungenauigkeit zu mildern bzw. zu beseitigen? (Begründung!)
- Für $\cos 887.5$ soll nun ein genauerer Wert in folgender Weise ermittelt werden: Man berechnet das reduzierte Argument r in erhöhter Genauigkeit (etwa mit 25 Dezimalstellen, mit Hilfe von MAPLE), übernimmt den sich daraus durch Rundung ergebenden Wert in das eigene Programm, und wertet dann cos aus.

  Man berechne die beiden Approximationen[3] für $\cos 887.5$, die sich mit diesen beiden Versionen von r ergeben ('normal' bzw. mit erhöhter Genauigkeit bestimmt), vergleiche diese mit dem 'exakten' Wert für $\cos 887.5$ und interpretiere die Resultate.
- Man bestimme auch die relative Konditionszahl der Auswertung von $\cos x$ für das 'kritische' Argument $x = 887.5$. Diese Konditionszahl ist groß (check!), aber da ja x exakt als Maschinenzahl gegeben ist (kein Datenfehler), erwartet man von einer guten Implementierung von cos, dass sie auch hier einen genauen Wert liefert.

(4.7*) Fleißaufgabe (nichttrivial!)

Unter (4.6) wurde das reduzierte Argument r für den konkreten Wert $x = 887.5$ mittels erhöhter Genauigkeit berechnet. Man versuche eine Methode zu finden und in MATLAB zu codieren, die für 'beliebige' (d.h. nicht allzu große) Argumente x (etwa $|x| \leq 10^5$) das zugehörige r möglichst genau berechnet (bis auf einen unvermeidlichen relativen Fehler der Größenordnung *eps*). (Grundlage dafür ist eine ausreichend genaue Approximation für 2π.)

[3] Um den Einfluss der verschiedenen Fehlerquellen sauber zu trennen, soll bei diesem Vergleich der Wert von $\cos r$ der Einfachheit halber mit der 'exakten' Standardfunktion cos ausgewertet werden, und nicht mit der selbstgebastelten Approximation.

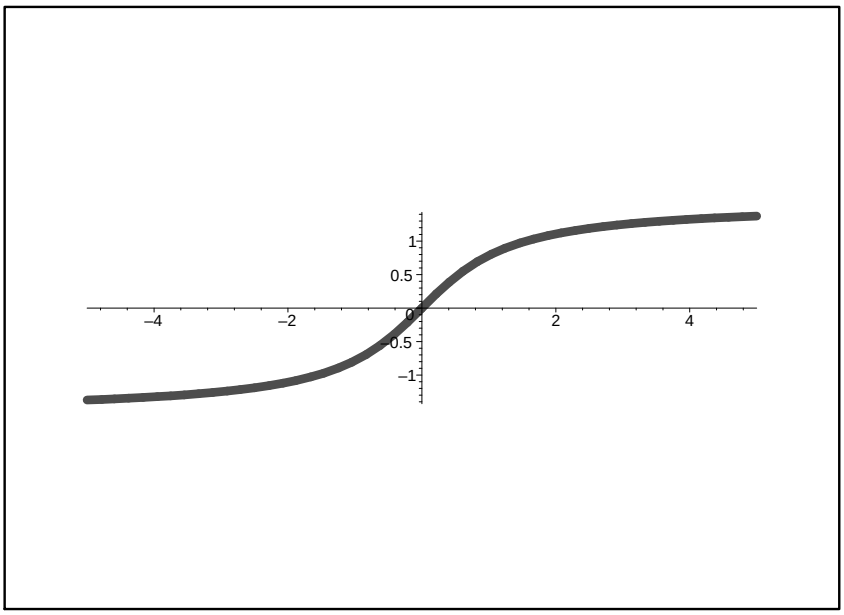

Abbildung 5: Arcustangens $\arctan x$

Beispiel 5: Arcustangens

Es soll eine Funktionsprozedur für die Arcustangensfunktion $\arctan(x)$ (Hauptzweig, mit Funktionswerten aus $(-\pi/2, \pi/2)$) entwickelt werden. Die Funktion wird dabei durch die ersten $n+1$ Summanden ihrer Taylorreihe bezüglich der Stelle $x_0 = 0$ ersetzt:

$$\arctan x \approx \sum_{i=0}^{n} (-1)^i \frac{x^{2i+1}}{2i+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \ldots + (-1)^n \frac{x^{2n+1}}{2n+1}$$

Diese Reihe konvergiert jedoch nur für $|x| \leq 1$. (Für $x = 1$ erhält man eine bekannte Reihenapproximation von $\pi/4$.)

(5.1) Man formuliere die Aufgabenstellung als numerisches Problem für einen konkreten Rechner:

- Genauigkeitsforderung: Es wird eine relative Genauigkeit von mindestens 100 *eps* angestrebt.
- Für welchen Bereich von x-Werten ist es (bei gegebener Arithmetik) überhaupt sinnvoll, eine Approximaton zu berechnen ('numerischer Definitionsbereich')? Welche Ausnahmesituation(en) liegt(liegen) außerhalb dieses Bereiches vor? Gibt es Symmetrien, so dass dieser Bereich weiter reduziert werden kann?
- Für welche speziellen Argumente $x \in \mathbb{F}$ ist der Funktionswert trivial?

(5.2) Wie viele Glieder der Taylorreihe muss man mindestens summieren, damit man die gewünschte Genauigkeit erreicht? Man führe eine Abschätzung für beliebige Argumente $x \in (0, 1]$ durch und stelle diese Information in geeigneter Weise grafisch dar.

Dabei wird exakte Rechnung vorausgesetzt: Die Rechenfehler, die beim Aufsummieren der Reihe entstehen, werden vernachlässigt. In diesem Fall gilt folgende Abschätzung für den Ver-

fahrensfehler:[4]

$$| \arctan x - \sum_{i=0}^{n} \frac{x^{2i+1}}{2i+1} | \leq \frac{x^{2n+3}}{2n+3}$$

(5.3) Für eine Folge von konkreten Argumente $x \in \mathbb{F}$,

$$x = 1 - 2^{-k}, \qquad k = 1, 2, \ldots$$

führe man die Summation dieser endlichen Reihen in natürlicher Reihenfolge durch und ermittle die relativen Fehler der so erhaltenen Resultate. (Als Bezugsgröße verwendet man den von der Standardprozedur gelieferten Wert.)

Wird eine relative Genauigkeit erreicht, die dem unter (5.2) angegebenen Verfahrensfehler entspricht? Wie steht es mit dem Rechenaufwand (d.h. wie viele Reihenglieder mussten summiert werden?)

(5.4) Man gebe eine Begründung für die beobachteten Phänomene und vergleiche die Qualität der obigen Resultate mit jener, die man bei Summation in umgekehrter Reihenfolge erhält. Ist ein Unterschied erkennbar?

(5.5) Man entwickle folgende praktikable Variante, die für Argumente x nahe an 1 eine schneller konvergente Reihe verwendet. Mit Hilfe von MAPLE erhält man (Taylorreihe an $x_0 = 1$):

$$\arctan(1+\delta) \approx \frac{\pi}{4} + \frac{\delta}{2} - \frac{\delta^2}{4} + \frac{\delta^3}{12} - \frac{\delta^5}{40} + \frac{\delta^6}{48} - \frac{\delta^7}{112} + \frac{\delta^9}{288} - \frac{\delta^{10}}{320} + \frac{\delta^{11}}{704} + \ldots$$

(Weitere Terme bzw. eine Restgliedabschätzung ermittle man bei Bedarf mittels MAPLE.) Im folgenden wird natürlich die verfügbare Approximation **pi** für π verwendet.

Man teste die numerische Konvergenz dieser Reihe für die gleiche Folge von Argumenten wie unter (5.2), und vergleiche insbesondere den Unterschied in der Genauigkeit bei umgekehrter Summationsreihenfolge, abhängig von der Anzahl der verwendeten Summanden.

(5.6) (nichttrivial:) In einer praktischen Implementierung besteht nun das Hauptproblem darin, wo man – bei gegebener Genauigkeitsforderung – zwischen den beiden obem verwendeten Reihenapproximationen (vgl. (5.2), (5.5)) 'umschalten' soll. man versuche sinnvolle Kriterien dafür zu definieren und setze eine entsprechende Strategie um.

Eine mögliche Alternative besteht darin, im Intervall $[0, 1]$ mit einer einzigen Taylorreihe zu arbeiten, etwa um den Punkt $x = 1/2$. Die Koeffizienten dieser Reihe kann man mit Hilfe von MAPLE bestimmen und in doppelter Genauigkeit übernehmen.

Schließlich implementiere und teste man folgende Approximation von $\arctan x$ für belienige x:

a) Ist $x \in [0, 1]$, so wird $\arctan x$ mit dem obigen Algorithmus ermittelt, wobei die jeweilige Taylorreihe in der umgekehrten Reihenfolge summiert wird. (Es empfiehlt sich ein zusätzlicger Summand als 'Genauigkeitsreserve'.)

b) Für $x > 1$ verwendet man die Identität

$$\arctan x = \frac{\pi}{2} - \arctan \frac{1}{x}.$$

c) Für $x < 0$ ist $\arctan x = -\arctan(-x)$.

[4] Bei einer konvergenten alternierenden Reihe kann man den Abbruchfehler durch den Betrag des ersten nicht mehr berücksichtigten Reihengliedes abschätzen.

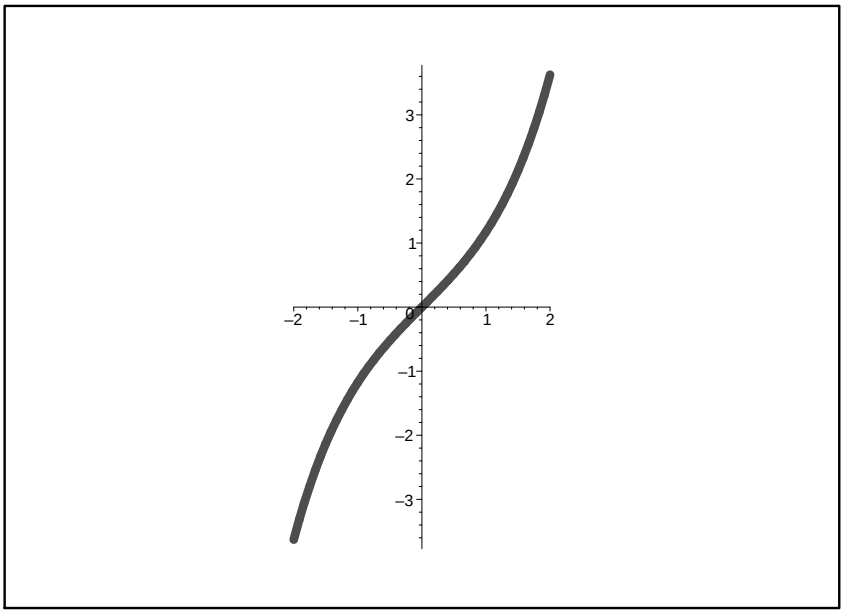

Abbildung 6: Sinus hyperbolicus $\sinh x$

Beispiel 6: Sinus hyperbolicus

Es soll eine Funktionsprozedur für den hyperbolischen Sinus $\sinh x$ entwickelt werden. Dazu werden zwei verschiedene Varianten in Betracht gezogen:

(i) Die Funktion wird durch die ersten $n+1$ Summanden ihrer Taylorreihe bezüglich der Stelle $x_0 = 0$ approximiert:

$$\sinh x \approx \sum_{i=0}^{n} \frac{x^{2i+1}}{(2i+1)!} = x + \frac{x^3}{6} + \frac{x^5}{120} + \ldots + \frac{x^{2n+1}}{(2n+1)!}$$

(ii) Unter der Annahme, dass eine Funktionsprozedur für die Exponentialfunktion $\exp(x) = e^x$ verfügbar ist, soll $\sinh x$ direkt auf Grund der Definition

$$\sinh x := \frac{e^x - e^{-x}}{2}$$

implementiert werden. Man hat also hier keinen Verfahrensfehler, aber natürlich einen elementaren Rundungsfehler bei der bei Auswertung der Exponentialfunktion.

(6.1) Man formuliere die Aufgabenstellung als numerisches Problem für einen konkreten Rechner:

- Genauigkeitsforderung: Es wird eine relative Genauigkeit von mindestens 100 *eps* angestrebt.
- Für welchen Bereich von x-Werten ist es (bei gegebener Arithmetik) überhaupt sinnvoll, eine Approximaton zu berechnen ('numerischer Definitionsbereich')? Welche Ausnahmesituation(en) liegt(liegen) außerhalb dieses Bereiches vor? Gibt es Symmetrien, so dass dieser Bereich weiter reduziert werden kann?
- Für welche speziellen Argumente $x \in \mathbb{F}$ ist der Funktionswert trivial?

(6.2) Wie viele Glieder der Taylorreihe muss man mindestens summieren, damit man die gewünschte Genauigkeit erreicht? Man führe eine Abschätzung für beliebige Argumente x aus dem numerischen Definitionsbereich durch und stelle diese Information in geeigneter Weise grafisch dar.

Dabei wird exakte Rechnung vorausgesetzt: Die Rechenfehler, die bei dem Aufsummieren der Reihe entstehen, werden vernachlässigt. Der Reihenrest,

$$\sinh x - \sum_{i=0}^{n} \frac{x^{2i+1}}{(2i+1)!} = \frac{x^{2n+3}}{(2n+3)!} \cosh(\theta x), \qquad \text{mit} \quad \theta \in [0,1],$$

entspricht dem absoluten Verfahrensfehler.

(6.3) Für eine Folge von konkreten Argumenten $x \in \mathbb{F}$,

$$x = 2^{\pm k}, \quad k = 0, 1, 2, 3, \ldots \quad \text{(so weit wie möglich)}$$

führe man die Summation dieser endlichen Reihen in natürlicher Reihenfolge durch und ermittle den Fehler der so erhaltenen Resultate. (Als Bezugsgröße verwendet man den von der Standardprozedur gelieferten Wert.)

Wird eine relative Genauigkeit erreicht, die dem unter (6.2) angegebenen Verfahrensfehler entspricht? Man gebe eine Begründung für die beobachteten Phänomene und vergleiche die Qualität der obigen Resultate mit jener, die man bei Summation in umgekehrter Reihenfolge erhält.

(6.4) Man versuche zu beurteilen, für welchen Bereich von x-Werten Variante (i) bzw. (ii) aus (6.2) der anderen vorzuziehen ist, und führe entsprechende numerische Tests durch. (Rechenaufwand und numerisch erzielte Genauigkeit sind die entscheidenden Kenngrößen.)

Was passiert für sehr betragskleine Argumente x? Welche Konsequenz ist also hinsichtlich einer konkreten Implementierung zu ziehen?

(6.5) Für die auf der Exponentialfunktion basierende Variante führe man eine formale Rundungsfehleranalyse durch. Dabei soll angenommen werden, dass $\exp(x)$ stets mit einem relativen Rundungsfehler der Größenordnung $10\,eps$ ausgewertet wird.

(6.6) Auf Grund der unter (6.4) und (6.5) gewonnenen Erkenntnisse implementiere und teste man schließlich folgende Approximation von $\sinh x$:

a) Je nach Größenordnung des Argumentes wird mit der entsprechend langen Taylorreihe oder mit der Exponentialfunktion gearbeitet (ein zusätzlicher Term als Genauigkeitsreserve). Der 'Umschaltpunkt' zwischen den beiden Varianten ist geeignet zu wählen (vgl. (6.5)).

b) Für $x < 0$ ist $\sinh x = -\sinh(-x)$.