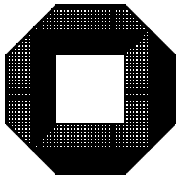


UNIVERSITÄT KARLSRUHE

# Algorithmen mit garantierten Fehlerschranken für die Fehler- und die komplementäre Fehlerfunktion

F. Blomquist und W. Krämer

Preprint Nr. 97/3

Institut für Wissenschaftliches Rechnen
und Mathematische Modellbildung


76128 Karlsruhe

**Anschrift der Verfasser:**

Dr. Frithjof Blomquist
Adlerweg 6
66346 Püttlingen

HDoz. Dr. Walter Krämer
Institut für Wissenschaftliches Rechnen und
Mathematische Modellbildung (IWRMM)

Universität Karlsruhe
Postfach 6980
76128 Karlsruhe Bundesrepublik Deutschland

Das Postscript-File dieses Preprints sowie die vollständigen Quelltexte aller Programme sind über FTP unter der Adresse
`iamk4515.mathematik.uni-karlsruhe.de`
in den Verzeichnissen
`/pub/iwrmm/preprints` bzw. `/pub/iwrmm/erf`
erhältlich.

# Algorithmen mit garantierten Fehlerschranken für die Fehler- und die komplementäre Fehlerfunktion

Frithjof Blomquist und Walter Krämer

## Inhaltsverzeichnis

# 1 Zusammenfassung

Die angegebenen Algorithmen erlauben die Berechnung von Einschließungen von Funktionswerten der Fehlerfunktion erf($x$) bzw. der komplementären Fehlerfunktion erfc($x$) für Punkt- und Intervallargumente, welche im IEEE-double Zahlformat [30] vorliegen. Sowohl die Approximationsfehler in den einzelnen Teilbereichen als auch alle auftretenden Rundungsfehler werden durch a priori Fehlerabschätzungen unter Einsatz von Intervallmethoden sicher erfaßt. Die erhaltenen worst-case Fehlerschranken für den maximalen relativen Fehler sind simultan für alle zulässigen Argumente gültig. Sie werden schließlich zur sicheren Einschließung von Wertebereichen über Punkten oder Intervallen verwendet.

Im Anhang findet sich eine vollständige XSC-Implementierung [26] der diskutierten Algorithmen. Alle Approximationskoeffizienten sind angegeben, so daß eine Übertragung in eine andere Programmiersprache sehr einfach ist.

**Key Words: Error Function, Complementary Error Function, Reliable Error Estimates, IEEE-double Format, Special Functions**

**MSC: 65D15, 65G05, 65G10, 68M15**

# 2 Einleitung

Für Algorithmen aus dem Bereich der Numerik mit Ergebnisverifikation werden sogenannte Intervallfunktionen benötigt, deren Ergebnisintervall den exakten Wertebereich der betrachteten Funktion über Intervallargumenten mit Sicherheit einschließt. Die berechnete Einschließung soll dabei möglichst eng sein.

Zur Implementierung solcher Funktionen sind sichere a-priori Abschätzungen der Approximationsfehler in den verschiedenen Teilbereichen sowie a-priori worst-case Fehlerabschätzungen der durch Rundungsfehler verursachten Ungenauigkeiten notwendig. In den ersten Abschnitten dieser Arbeit wird diese Thematik zunächst generell diskutiert. XSC-Programme, die es erlauben, große Teile der Fehlerabschätzungen automatisch mit dem Rechner durchzuführen, sind in [3, 4, 10, 11, 14, 16] näher beschrieben.

Im eigentlichen Hauptabschnitt werden die Algorithmen für die Fehlerfunktion erf($x$) und deren Komplement erfc($x$) für Punktargumente angegeben. Für die einzelnen Teilbereiche werden die Hilfsapproximationsfunktionen hergeleitet und deren Approximationsfehler analytisch bestimmt. Mit Hilfe dieser (programmierbaren und nahezu perfekten) Hilfsnäherungen können dann die Approximationsfehler der später tatsächlich zum Einsatz kommenden rationalen Approximationen automatisch durch die Verwendung von numerisch sicheren Intervallmethoden mit dem Rechner bestimmt werden.

Die gefundenen relativen worst-case Fehlerschranken werden schließlich dazu verwendet, die Intervallroutinen aus den Punktfunktionen zusammenzusetzen. Durch die Berechnung von Einschließungen können hier die Sonderbehandlungen für Argumente entfallen, die auf Funktionswerte im Unterlaufbereich führen (für solche

Argumente gilt die gefundene relative Fehlerschranke i. a. nicht). Diese Funktionswerte werden entweder auf 0 oder auf die kleinste positive oder auf die größte negative normalisierte Gleitkommazahl abgebildet, eben gerade so, daß die gewünschte Einschließungseigenschaft gewährleistet bleibt.

Die vollständigen XSC-Programmlistings sowohl der Fehlerfunktion als auch der komplementären Fehlerfunktion für Punkt- und Intervallargumente sowie der Quellcode einiger wichtiger Hilfsroutinen sind über FTP unter der Adresse `iamk4515.mathematik.uni-karlsruhe.de` im Verzeichnis `/pub/iwrmm/erf` erhältlich. Weiter ist im Anhang dieser Arbeit ein kurzes Testprogramm mit numerischen Ergebnissen angegeben.

# 3 Zur Notation

| | |
|---|---|
| $S = S(b,l) = S(b,l,\underline{e},\overline{e})$ | Gleitkommaraster mit Basis $b$, Mantissenlänge $l$ und Exponent $e$ mit $\underline{e} \le e \le \overline{e}$ |
| $S(2,53,-1022,1023)$ | IEEE-Datenformat double |
| $\circ \in \{+,-,\bullet,/\}$ | exakte reelle Operation |
| $\boxdot, \circ \in \{+,-,\bullet,/\}$ | Gleitkommaoperator, Maschinenoperation mit Rundung zur nächstgelegenen Gleitkommazahl |
| $\triangledown, \circ \in \{+,-,\bullet,/\}$ | Maschinenoperation mit Rundung nach unten |
| $\triangle, \circ \in \{+,-,\bullet,/\}$ | Maschinenoperation mit Rundung nach oben |
| $\lvert a \circ b - a \boxdot b\rvert := \lvert(a \circ b) - (a \boxdot b)\rvert$ | Implizite Klammerung beachten! |
| `MinReal` | kleinste positive normalisierte Gleitkommazahl, für IEEE-Datenformat gilt: `MinReal`:=$2.22\ldots \cdot 10^{-308}$ |
| `MaxReal` | größte normalisierte Gleitkommazahl, für IEEE-Datenformat gilt: `MaxReal`:=$1.78\ldots \cdot 10^{308}$ |
| $a, x, f, \ldots$ | exakte Größen |
| $\tilde{a}, \tilde{x}, \tilde{f}, \ldots$ | auf der Maschine berechnete, i.a. fehlerbehaftete Größen |
| ulp | eine Einheit in der letzten Mantissenstelle (unit last place) |
| $\varepsilon^* := \frac{1}{2}2^{1-l} = 2^{-l}$ | relative Maschinengenauigkeit bzgl. $S(b,l)$ |
| `eps52` | `eps52`:= $2^{1-53} = 2.22044\ldots \cdot 10^{-16}$ |
| `eps53` | `eps53`:= $\frac{1}{2}2^{1-53} = 1.11022\ldots \cdot 10^{-16} = \varepsilon^*$ |
| $\mathrm{succ}(x), x \in S$ | Gleitkommanachfolger von $x$ |
| $\mathbb{R}^+$ | Menge der positiven reellen Zahlen |
| $I\mathbb{R}$ | Menge der abgeschlossenen Intervalle über den reellen Zahlen |
| $X = [\underline{x}, \overline{x}] \in I\mathbb{R}$ | Notation für Intervalle |
| $IS := \{[\underline{a},\overline{a}] \mid \underline{a},\overline{a} \in S, \underline{a} \le \overline{a}\}$ | Menge der Maschinenintervalle |
| $\lvert A\rvert, A \in I\mathbb{R}$ | $\lvert A\rvert := \max_{a \in A} \lvert a\rvert$, Betragsmaximum |

| | |
|---|---|
| $\langle A \rangle, A \in I\!I\!R$ | $\langle A \rangle := \min\limits_{a \in A} \lvert a \rvert$, Betragsminimum |
| $\mathrm{diam}(A) := \sup(A) - \inf(A)$ | Durchmesser eines Intervalls $A \in I\!I\!R$ |
| $W_f(X) := \{f(x) \mid x \in X\}$ | Wertebereich von $f$ über Intervall $X$ |
| $H(x) \approx f(x)$ | nahezu perfekte Hilfsapproximationsfunktion an die zu approximierende Funktion $f(x)$. $H(x)$ hängt in der Regel auch vom aktuell betrachteten Approximationsbereich ab. $H(x)$ dient der automatischen Fehlerabschätzung für die eigentlich gesuchte und später implementierte effiziente Approximation. |
| $\varepsilon(\mathrm{app}, 1)$ | relative Fehlerschranke der nahezu perfekten Approximation $H(x)$ an $f$ |
| $\varepsilon(\mathrm{app}, 2)$ | relative Fehlerschranke der implementierten Approximation (bezogen auf $H(x)$) |
| $\varepsilon(f)$ | relative Gesamtfehlerschranke der endgültigen Maschinenrealisierung $\widetilde{f}$ von $f$ |

# 4 Generelles zur Fehleraufspaltung

In diesem Abschnitt werden diejenigen Fehler berücksichtigt, die bei der Auswertung einer vorgegebenen stetigen Funktion

$$h \,:\, [c,d\,] \longrightarrow I\!R, \quad c, d \in S(B,k)$$

auf einem Rechner mit Gleitkommasystem $S$ mit $k$ Mantissenziffern zur Basis $B$ üblicherweise auftreten. Die möglichen Fehler entstehen bei der:

1. Berechnung eines reduzierten Arguments $x$ bzw. dessen Maschinennäherung $\widetilde{x}$ mit Hilfe der stetigen Funktion:

$$r \,:\, [c,d\,] \longrightarrow I\!R, \qquad W_r([c,d\,]) = [a,b] := \{\, x \mid x = r(t),\, t \in [c,d\,]\,\}$$

$$\widetilde{x} = \tilde{r}(t) = x \cdot (1 + \varepsilon_x), \qquad |\varepsilon_x| \le \varepsilon(x) \quad \text{für alle } x \in [a,b], \quad a, b \in I\!R.$$

2. Approximation von $h(t) = f(r(t)) = f(x)$ durch $g(x)$:

$$f(x) \approx g(x), \quad x = r(t) \in [a,b];$$

$$\varepsilon_{\mathrm{app}} := \frac{g(x) - f(x)}{f(x)}, \qquad |\varepsilon_{\mathrm{app}}| \le \varepsilon(\mathrm{app}) \quad \text{für alle } x \in [a,b] \text{ mit } f(x) \neq 0.$$

3. Auswertung der Approximationsfunktion $g$ für die durch Argumentreduktion i.a. gestörten $\widetilde{x}$-Werte:

$$\varepsilon_g := \frac{\widetilde{g}(\widetilde{x}) - g(x)}{g(x)}, \qquad |\varepsilon_g| \le \varepsilon(g), \quad \widetilde{x} = x \cdot (1 + \varepsilon_x), \quad x \in [a,b].$$

Wird also für $t \in [c,d]$ mit Hilfe der stetigen Funktion $x = r(t)$ ein reduziertes Argument $x$ berechnet, und ist $[a,b]$ mit $a,b \in I\!R$ der Wertebereich von $r$, so ist an Stelle von $h(t)$ mit $t \in [c,d]$ die Funktion $f(x) = f(r(t)) = h(t)$ mit $x \in [a,b]$ auszuwerten. Da $x = r(t)$ zu vorgegebenem $t$ auf der Maschine berechnet wird, erhält man nicht $x \in [a,b]$, sondern den i. a. fehlerbehafteten Wert $\tilde{x} = x \cdot (1 + \varepsilon_x) \in S$ mit der relativen Fehlerschranke $\varepsilon(x)$. Ist $g(x) \approx f(x)$ für $x \in [a,b]$ die Approximationsfunktion, so wird diese also nicht mit dem exakten $x$, sondern mit $\tilde{x} = x \cdot (1 + \varepsilon_k) \in S(B,k)$ ausgewertet. Da bei dieser Auswertung auf dem Rechner weitere Rundungsfehler zu erwarten sind, erhält man nicht $g(\tilde{x}) \in I\!R$, sondern den i. a. fehlerbehafteten Maschinenwert $\tilde{g}(\tilde{x}) \in S(B,k)$. Statt $f(x) = h(t)$ erhält man also nur $\tilde{g}(\tilde{x})$ mit dem relativen Fehler

$$\varepsilon_h(t) := \frac{\tilde{g}(\tilde{x}) - h(t)}{h(t)} = \frac{\tilde{g}(\tilde{x}) - f(x)}{f(x)},$$

wobei $f(x) \neq 0$ für $x \in [a,b]$ vorausgesetzt wird. Mit Hilfe der Fehleraufspaltung läßt sich nun $|\varepsilon_h(t)|$ wie folgt abschätzen [12]:

$$(1) \qquad \begin{aligned} \left|\frac{f(x) - \tilde{g}(\tilde{x})}{f(x)}\right| &= \left|\frac{f(x) - g(x) + g(x) - \tilde{g}(\tilde{x})}{f(x)}\right| \\ &\leq \left|\frac{f(x) - g(x)}{f(x)}\right| + \left|\frac{g(x) - \tilde{g}(\tilde{x})}{g(x)}\right| \cdot \left|\frac{g(x) - f(x) + f(x)}{f(x)}\right| \\ &\leq \varepsilon(\text{app}) + [1 + \varepsilon(\text{app})] \cdot \varepsilon(g) =: \varepsilon(h) \end{aligned}$$

$\varepsilon(\text{app})$ ist der relative Approximationsfehler und $\varepsilon(g)$ bedeutet den relativen Auswertefehler der Approximationsfunktion $g$. $\varepsilon(h)$ ist die Gesamtfehlerschranke von $h(t)$ für $t \in [c,d]$, wenn das reduzierte Argumente $x = r(t)$ nur näherungsweise und damit i. a. fehlerbehaftet zu $\tilde{x} = x \cdot (1 + \varepsilon_x), \quad |\varepsilon_x| \leq \varepsilon(x)$ berechnet wird, und wenn die Approximationsfunktion $g$ anschließend für dieses gestörte Argument $\tilde{x}$ unter Verwendung von Gleitkommaoperationen auf der Maschine ausgewertet wird: $\tilde{g}(\tilde{x}) = g(x) \cdot (1 + \varepsilon_g); \quad |\varepsilon_g| \leq \varepsilon(g)$.

Berechnet man $\varepsilon(h)$ gemäß Formel (1), so ist sichergestellt, daß

$$\left|\frac{h(t) - \tilde{g}(\tilde{r}(t))}{h(t)}\right| \leq \varepsilon(h), \qquad \text{simultan für alle } t \in [c,d]$$

gilt. Dies bedeutet, daß $\varepsilon(h)$ gerade die gesuchte verläßliche Gesamtfehlerschranke für die Maschinenrealisierung $\tilde{h}(t) := \tilde{g}(\tilde{r}(t))$ von $h(t)$ und damit von $f(x)$ darstellt.

# 5 Prinzipielles zu Fehlerabschätzungen bei speziellen Funktionen

Im Zusammenhang mit der Realisierung spezieller Funktionen (Fehlerfunktion, Gammafunktion, Besselfunktionen, ... ) in Rechenanlagen zeigt es sich, daß es für die sichere Fehlerabschätzung oft sinnvoll ist, zusätzlich zur eigentlich gesuchten, auf dem Rechner zu realisierenden Näherung $g(x)$, eine Hilfsnäherungsfunktion $H(x)$ (diese ist in der Regel eine fast perfekte, z. B. mit einer Langzahlarithmetik berechenbare Approximation) zu verwenden. Genauer kann man oft wie im folgenden dargestellt verfahren.

Es wird wieder von einer stetigen und reellwertigen Funktion

$$f \,:\, [a,b] \longrightarrow I\!R$$

ausgegangen, die auf einem Rechner mit Gleitkommaraster $S(B,k)$ ausgewertet werden soll. Um kurze Laufzeiten zu erhalten, soll die Approximationsfunktion $g \approx f$ möglichst einfach (z. B. rationale Funktion) aufgebaut sein. Bezeichnet man wieder mit $\widetilde{f}(x)$ das i.a. fehlerbehaftete Maschinenergebnis, so gilt

$$\widetilde{f}(x) = f(x)(1+\varepsilon_f), \qquad |\varepsilon_f| \le \varepsilon(f) \quad \text{für alle } x \in [a,b] \cap S(B,k).$$

Zur Berechnung der gesuchten Fehlerschranke $\varepsilon(f)$ sind i.a. **zwei** Approximationsschritte notwendig. Zunächst muß eine Hilfsfunktion $H(x)$ als Approximation an $f(x)$ gefunden werden, welche durchaus recht kompliziert aufgebaut sein darf. Es wird nur verlangt, daß $H(x)$ mit Hilfe von bereits implementierten Intervallfunktionen bzw. Intervalloperationen programmierbar ist. Die zugehörige Approximationsfehlerschranke $\varepsilon(\text{app},1)$ muß in der Regel analytisch (per Hand) hergeleitet werden. Die Gewinnung der eigentlichen Approximationsfunktion $g(x)$ für die Implementierung der Ausgangsfunktion $f(x)$ geschieht dann mit Hilfe von $H(x)$. Eine Schranke $\varepsilon(\text{app},2)$ für den hierbei auftretenden (zweiten) Approximationsfehler kann nun mit intervallarithmetischen Mitteln automatisch bestimmt werden. Genauer wird wie folgt vorgegangen:

**Schritt 1:** $\qquad f(x) \approx H(x), \qquad x \in [a,b],$

wobei $H(x)$ nur aus den Standardfunktionen aufgebaut sein soll, die das Intervall-Langzahlmodul `mpitaylor` zur Verfügung stellt. Der relative Approximationsfehler ist für $f(x) \neq 0$ gegeben durch:

$$\varepsilon_{\text{app},1} = \frac{f(x) - H(x)}{f(x)}; \qquad |\varepsilon_{\text{app},1}| \le \varepsilon(\text{app},1), \quad x \in [a,b].$$

Die Berechnung von $\varepsilon(\text{app},1)$ ist somit ein rein mathematisches Problem, das für jede Funktion individuell zu lösen ist. Dabei wird nicht verlangt, daß $H(x)$ auf dem Rechner möglichst schnell auszuwerten ist; das Hauptziel ist vielmehr die Berechnung einer **garantierten** Oberschranke $\varepsilon(\text{app},1)$ des entsprechenden Approximationsfehlers!

**Schritt 2:** $H(x) \approx g(x), \qquad x \in [a,b],$
wobei $g$ jetzt diejenige Funktion bezeichnet, mit der $f$ auf dem Rechner tatsächlich approximiert wird. Um kurze Laufzeiten zu erhalten, wird $g$ in vielen Fällen als gebrochen rationale Funktion, deren Koeffizienten z.B. mit Hilfe eines Computeralgebrasystems berechnet werden können, gewählt. Für den relativen Approximationsfehler (bezüglich der Hilfsfunktion $H(x)$) kann man unter der Annahme $H(x) \neq 0$ die Oberschranke $\varepsilon(\mathrm{app},2)$ automatisch mit dem Rechner bestimmen:

$$\varepsilon_{\mathrm{app},2} = \frac{H(x)-g(x)}{H(x)}, \qquad |\varepsilon_{\mathrm{app},2}| \leq \varepsilon(\mathrm{app},2), \quad x \in [a,b].$$

Hierzu kann das XSC-Programm `AppErr` [4, 14] verwendet werden.

Bezüglich der Näherung $f \approx g$ ist der relative Approximationsfehler definiert durch

$$\varepsilon_{\mathrm{app}} := \frac{f(x)-g(x)}{f(x)}, \qquad |\varepsilon_{\mathrm{app}}| \leq \varepsilon(\mathrm{app}), \quad x \in [a,b]$$

und $|\varepsilon_{\mathrm{app}}|$ läßt sich mit Hilfe der Schranken $\varepsilon(\mathrm{app},1)$ und $\varepsilon(\mathrm{app},2)$ durch Anwendung der Dreiecksungleichung abschätzen:

(2) $$\boxed{|\varepsilon_{\mathrm{app}}| \leq \varepsilon(\mathrm{app},1) + [1+\varepsilon(\mathrm{app},1)] \cdot \varepsilon(\mathrm{app},2) =: \varepsilon(\mathrm{app})\,.}$$

Bezeichnet man mit $\tilde{g}(x)$ das i.a. fehlerbehaftete Rechnerergebnis von $g$, so ergibt sich die Darstellung

$$\tilde{g}(x) = g \cdot (1+\varepsilon_g)\,,$$

und die Oberschranke $\varepsilon(g)$ für den maximalen Betrag des relativen Auswertefehlers $\varepsilon_g$ läßt sich mit Hilfe eines XSC-Programms [4, 10] automatisch berechnen. Danach gilt dann

$$\tilde{g}(x) = g(x) \cdot (1+\varepsilon_g); \qquad |\varepsilon_g| \leq \varepsilon(g) \quad \text{für alle } x \in [a,b] \cap S(B,k),$$

Mit der Gesamtfehlerabschätzung (1) erhält man schließlich mit $\tilde{f}(x) = f(x)(1+\varepsilon_f)$

(3) $$\boxed{|\varepsilon_f| \leq \varepsilon(\mathrm{app}) + [1+\varepsilon(\mathrm{app})] \cdot \varepsilon(g) =: \varepsilon(f)\,.}$$

Vorausgesetzt ist dabei

$$x \in [a,b] \cap S(B,k) \ \wedge \ f = 0 \quad \Longrightarrow \quad \tilde{f}(x) \equiv \tilde{g}(x) = 0\,.$$

Diese Forderung kann durch eine geeignete Sonderbehandlung der Nullstellen der betrachteten Funktion bei der Entwicklung des Algorithmus für $g$ in der Regel einfach erfüllt werden.

# 6 Zum Approximationsfehler

## 6.1 Übersicht

Bei der Implementierung einer über einem reellen Intervall definierten reellwertigen Funktion

$$f : [a,b] \longrightarrow I\!R$$

auf einem Rechner sind die folgenden zwei Punkte zu beachten:

- Um möglichst kurze Laufzeiten zu erhalten, sollte $f$ über dem Intervall $[a,b]$ durch eine rationale Funktion approximiert werden, wobei Zähler und Nenner durch je ein Polynom definiert sind:

$$f(x) \approx g(x) := \frac{\mathrm{P}_N(x)}{\mathrm{Q}_M(x)}; \quad \mathrm{Q}_M(x) \neq 0, \quad x \in [a,b], \quad N, M \in \{0, 1, 2, \ \ldots\}$$

- Wegen der Approximation $f(x) \approx g(x)$ und wegen der i.a. unvermeidbaren Rundungsfehler bei der Auswertung von $g$ wird das Maschinenergebnis mit dem exakten Funktionswert $f(x)$ nur in Ausnahmefällen übereinstimmen. Zur Berechnung einer **garantierten** Fehlerschranke ist es daher u.a. notwendig, eine ebenfalls garantierte Oberschranke für den Approximationsfehler für alle $x \in [a,b]$ zu bestimmen.

Der absolute bzw. relative Approximationsfehler ist definiert durch

$$\begin{aligned} \Delta(x) &:= f(x) - g(x), \quad |\Delta(x)| \leq \Delta(\mathrm{app}), \quad x \in [a,b], \\ \varepsilon(x) &:= \frac{f(x) - g(x)}{f(x)}, \quad |\varepsilon(x)| \leq \varepsilon(\mathrm{app}), \quad f(x) \neq 0, \quad x \in [a,b]. \end{aligned}$$

Zur Bestimmung der Oberschranken $\Delta(\mathrm{app}), \varepsilon(\mathrm{app})$ gibt es in Abhängigkeit von der vorgegebenen Funktion $f$ und ihrem Approximationsintervall $[a,b]$ verschiedene Methoden. Dabei dürfen $\Delta(\mathrm{app}), \varepsilon(\mathrm{app})$ natürlich nicht zu groß ausfallen, und vom mathematischen Standpunkt aus müssen die Oberschranken **korrekt** berechnet werden.

Bei den Standardfunktionen ( $f$ = exp, sin, arctan, ... ) läßt sich $[a,b]$ auf ein relativ schmales Intervall reduzieren, dessen Mittelpunkt der Koordinatenursprung ist. Als Approximationsfunktion kann daher ein Taylor-Polynom niedriger Ordnung (kurze Laufzeit) gewählt werden, und der Rest der Taylorreihe läßt sich entweder durch eine geometrische Reihe oder durch das nachfolgende Reihenglied abschätzen, wenn die Potenzreihe eine Leibnizreihe ist [6, 12]. Dadurch erhält man für die Oberschranke des Approximationsfehlers einen in geschlossener Form vorliegenden einfachen Ausdruck, der durch Intervallrechnung sicher nach oben abgeschätzt werden kann.

Bei der Berechnung des Approximationsfehlers für die speziellen Funktionen der mathematischen Physik liegen die Dinge etwas anders, da jetzt im Gegensatz zu

den Standardfunktionen eine Argumentreduktion auf ein sehr schmales Intervall i. a. nicht möglich ist.

Das folgende Beispiel soll den Sachverhalt verdeutlichen. Mit der Riemannschen Zeta-Funktion $\zeta(x)$ und der Eulerschen Konstanten $\gamma = 0.57721\ldots$ gilt für die Funktion $f(x) = -\ln(\Gamma(x))$ die Reihenentwicklung

$$f(x) \equiv (x-2)(\gamma-1) - \sum_{k=2}^{\infty}(-1)^k[\zeta(k)-1]\cdot\frac{(x-2)^k}{k}, \quad |x-2| \le \frac{1}{2}.$$

Approximiert man nun $f(x)$ im Intervall $[1.5, 2.5]$ durch das Taylorpolynom

$$\mathrm{T}_N(x) := (x-2)(\gamma-1) - \sum_{k=2}^{N}(-1)^k[\zeta(k)-1]\cdot\frac{(x-2)^k}{k} \approx f(x),$$

so erhält man nach dem oben beschriebenen Verfahren erst mit $N$=26 für den absoluten Fehler die Oberschranke $\Delta(\mathrm{app}) = 4.1121\cdot 10^{-18}$. Aus Laufzeitgründen ist die Rechnerauswertung von $\mathrm{T}_{26}(x)$ also völlig unakzeptabel. Die Rechenzeit reduziert sich jedoch etwa um den Faktor 2.4, wenn man $\mathrm{T}_{26}(x)$ durch eine rationale Bestapproximation ersetzt

$$f(x) \approx \mathrm{T}_{26}(x) \approx \frac{\mathrm{P}_6(x-2)}{\mathrm{Q}_5(x-2)}, \quad |x-2| \le \frac{1}{2} \tag{4}$$

mit

$$\mathrm{P}_6(x-2) := \sum_{k=0}^{6} a_k\cdot(x-2)^k, \qquad \mathrm{Q}_5(x-2) := \sum_{k=0}^{5} b_k\cdot(x-2)^k\,.$$

Die Polynomkoeffizienten $a_k, b_k$ können dabei z.B. mit einem Computeralgebrasystem (Langzahlrechnung) bestimmt werden. Für die rationale Approximation muß jetzt der Approximationsfehler bzgl. $T_{26}$ sicher abgeschätzt werden.
Um dieses Problem etwas allgemeiner zu formulieren, ersetzt man das spezielle Polynom $\mathrm{T}_{26}(x)$ durch eine hinreichend oft differenzierbare Hilfsfunktion $H(x)$, welche im Bereich $|x-x_0| \le \eta$ nur aus endlich vielen Standardfunktionen aufgebaut sein soll, die im XSC-Modul `mpitaylor` bereitgestellt werden. Man sucht also für die folgende Approximation

$$H(x) \approx \frac{\mathrm{P}_N(x-x_0)}{\mathrm{Q}_M(x-x_0)}, \quad \mathrm{Q}_M(x-x_0) \neq 0, \quad |x-x_0| \le \eta \tag{5}$$

die Oberschranken $\Delta(\mathrm{app})$ bzw. $\varepsilon(\mathrm{app})$ des absoluten bzw. relativen Approximationsfehlers. Für $\varepsilon(x)$ ergibt sich z. B. die Darstellung

$$\varepsilon(x) := \frac{\mathrm{P}_N(x-x_0) - \mathrm{Q}_M(x-x_0)\cdot H(x)}{\mathrm{Q}_M(x-x_0)\cdot H(x)}, \quad |x-x_0| \le \eta, \ \text{Nenner} \neq 0, \tag{6}$$

und die Berechnung einer garantierten Oberschranke $\varepsilon(\mathrm{app})$ für $|\varepsilon(x)|$ erscheint auf den ersten Blick einfach, da man gewohnt ist, ähnliche Probleme mit Werkzeugen der (verifizierten) globalen Optimierung ohne Schwierigkeiten zu lösen. Es stellt sich

heraus, daß $\varepsilon(\mathrm{app})$ durch globale Optimierung grundsätzlich nicht bestimmt werden kann! Um dies einzusehen, sei zunächst daran erinnert, daß die globale Optimierung nur dann zum Ziel führt, wenn man die Wertebereiche von $\varepsilon(x)$ für jedes Teilintervall einer endlichen Zerlegung von $|x - x_0| \leq \eta$ **ohne** wesentliche Überschätzung berechnen kann [9, S.106]. Bedeutet $[x]$ ein solches Teilintervall, so ist im Zähler von (6) der Ausdruck

$$\mathrm{P}_N([x] - x_0) - \mathrm{Q}_M([x] - x_0) \cdot H([x]) \tag{7}$$

intervallmäßig auszuwerten, wobei die Ergebnisintervalle von Minuend und Subtrahend um so besser übereinstimmen, je höher bei der rationalen Approximation die Polynomgrade $N, M$ gewählt werden. Im Ausdruck (7) sind damit zwei fast identische und nicht punktförmige Intervalle zu subtrahieren, was bekanntlich zu sehr starken Überschätzungen führt. Dies ist der Grund für das Versagen der globalen Optimierungsalgorithmen beim Untersuchen von Fehlerkurven bei Approximationsproblemen. Das Problem wird übrigens auch nicht dadurch gelöst, daß man in (7) die beiden Intervall-Summanden mit dem Intervall-Langzahlmodul `mpi_ari` [13] in hoher Genauigkeit auswertet, denn die Tatsache, daß zwei fast identische Intervalle zu subtrahieren sind, wird auch durch eine Langzahlarithmetik nicht beseitigt. Eine rein theoretische Lösung würde darin bestehen, daß man die Teilintervalle $[x]$ quasi punktförmig wählt, was jedoch auf völlig unpraktikable Rechenzeiten führen würde.

Eine Abschätzung des Approximationsfehlers wird also nur möglich sein, wenn es gelingt, den Ausdruck (7) so umzuformen, daß die Subtraktion fast identischer Intervalle vermieden wird. Dazu entwickeln man $H(x)$ im Punkt $x_0$ z. B. mit den Methoden der automatischen Differentiation in ein Taylor-Polynom mit Restglied $H(x) = \sum_{k=0}^{K} s_k \cdot (x - x_0)^k + \mathrm{R}(x, K)$ . Man erhält für den Ausdruck (7) die Darstellung

$$\underbrace{\left[\mathrm{P}_N(x - x_0) - \mathrm{Q}_M(x - x_0) \cdot \sum_{k=0}^{K} s_k \cdot (x - x_0)^k\right]}_{(*)} - \mathrm{Q}_M(x - x_0) \cdot \mathrm{R}(x, K)\,. \tag{8}$$

Der eigentliche Trick besteht nun darin, die Polynomdifferenz $(*)$ **direkt** zu berechnen, indem man in (8) zunächst die Koeffizienten des Subtrahenden bestimmt und anschließend die Differenz der entsprechenden Polynomkoeffizienten bildet. Schließt man nämlich die Koeffizienten von Minuend und Subtrahend mit einer Langzahl-Intervallarithmetik ein, so erhält man quasi punktförmige Intervalle, die so weit getrennt liegen, daß ihre Differenzen nahezu ohne Überschätzungen berechnet werden können! Damit ist die Auswertung von $(*)$ auf die Auswertung nur eines Polynoms zurückgeführt, welche z.B. nach dem Intervall-Hornerschema geschehen kann, wenn man $|x - x_0| \leq \eta$ im Bedarfsfall zur Vermeidung von Überschätzungen beim Hornerschema noch in mehrere Teilintervalle unterteilt. Die Abschätzung des Restgliedes in der Lagrangeschen Form erfolgt durch automatische Differentiation und Auswertung der $(K + 1)$-ten Ableitung von $H(x)$ über dem Intervall $|x - x_0| \leq \eta$, wobei eine Intervallunterteilung ebenfalls notwendig werden kann.

Es sei betont, daß das beschriebene Verfahren sehr deutlich zeigt, daß die naive Anwendung der Intervallrechnung nicht zum gewünschten Ziel führt, während beim

gezielten Einsatz an der richtigen Stelle (Differenz der Polynom-Koeffizienten) die Intervallarithmetik ein äußerst nützliches Werkzeug ist!

## 6.2 Rationale Approximation

In diesem Abschnitt betrachten wir die Approximation einer vorgegebenen Hilfsfunktion $H(x)$ durch eine rationale Funktion und zeigen, wie eine garantierte Oberschranke des absoluten bzw. relativen Approximationsfehlers berechnet werden kann. Mit Hilfe eines XSC-Programms lassen sich diese Schranken automatisch berechnen.

Da das Lösungsverfahren die automatische Differentiation benötigt, wird vorausgesetzt, daß die Hilfsfunktion $H(x)$ als ein endlicher Ausdruck in den Funktionen

$$\text{exp, ln, sqr, sqrt, sin, cos, arctan, pow}$$

sowie den Grundoperationen $-(\text{unär}), +, -, *, /$ gegeben ist. (Solche Ausdrücke können derzeit im Modul `mpitaylor` bearbeitet werden.)

Die rationale Funktion zur Approximation von $H(x)$ möge

$$A_0 + A_1 \cdot (x - x_0)^1 + \ \ldots \ + A_N \cdot (x - x_0)^N$$

als Zählerpolynom und

$$B_0 + B_1 \cdot (x - x_0)^1 + \ \ldots \ + B_M \cdot (x - x_0)^M$$

als Nennerpolynom besitzen. Bei festen Polynomgraden $N, M$ erhält man häufig eine sehr effektive Approximation, wenn die Koeffizienten $A_j, B_j$ nach Tschebyscheff bestimmt werden. Der absolute oder relative Approximationsfehler besitzt dann im Innern des Approximationsintervalls $|x - x_0| \leq \eta$ mindestens $N + M$ relative Extremstellen mit oszillierenden aber betragsgleichen Extremwerten. Die Betragsgleichheit der Extremwerte kann jedoch in der Praxis aus folgenden Gründen nicht realisiert werden:

- Die $A_j, B_j$ können nur mit endlich vielen Dezimalstellen berechnet werden.
- Es ist i. a. sinnvoll, die $A_j, B_j$ zur nächstgelegenen Zahl in dem Raster zu runden, in dem die Polynome ausgewertet werden sollen.

Durch die notwendige Rundung der Polynomkoeffizienten werden die absoluten Extremwerte des Approximationsfehlers verschieden sein, und man wird i.a. auch nicht garantieren können, daß sich die Anzahl der Extremstellen nicht ändert.

Bezeichnet man die aus den $A_j, B_j$ durch geeignete Rundung hervorgegangenen Koeffizienten mit $a_j, b_j$ , so lauten die tatsächlich für die Approximation auf der Maschine verwendeten Polynome

$$\begin{aligned} \mathrm{P}_N(x - x_0) &:= a_0 + a_1 \cdot (x - x_0)^1 + \ \ldots \ + a_N \cdot (x - x_0)^N; \quad \text{Zählerpolynom} \\ \mathrm{Q}_M(x - x_0) &:= b_0 + b_1 \cdot (x - x_0)^1 + \ \ldots \ + b_M \cdot (x - x_0)^M; \quad \text{Nennerpolynom} \end{aligned}$$

d. h. $H(x)$ wird durch

$$H(x) \approx \frac{\mathrm{P}_N(x-x_0)}{\mathrm{Q}_M(x-x_0)}, \quad \mathrm{Q}_M(x-x_0) \neq 0, \quad |x-x_0| \leq \eta$$

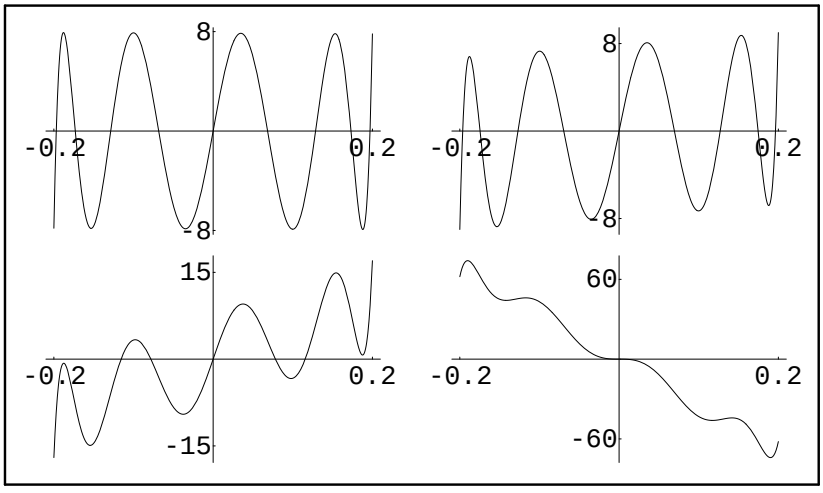
approximiert.

Das folgende Beispiel zeigt an Hand von vier Graphen den Einfluß der Rundung der Polynomkoeffizienten auf den Funktionsverlauf des relativen Approximationsfehlers.
Es wird $H(x) := e^x, \quad x_0 = 0, \quad \eta = 0.2, \quad M = N = 4$; betrachtet. Die mit Mathematica berechneten Koeffizienten $A_j, B_j$ lauten mit den ersten 18 Dezimalstellen

| $j$ | $A_j$ | $B_j$ |
|---|---|---|
| 0 | $9.99999999999999\ldots \cdot 10^{-1}$ | $+1.00000000000000\ldots \cdot 10^{+0}$ |
| 1 | $4.99999999999998228\ldots \cdot 10^{-1}$ | $-4.99999999999998228\ldots \cdot 10^{-1}$ |
| 2 | $1.07140305127468128\ldots \cdot 10^{-1}$ | $+1.07140305127468128\ldots \cdot 10^{-1}$ |
| 3 | $1.19034858976579290\ldots \cdot 10^{-2}$ | $-1.19034858976579290\ldots \cdot 10^{-2}$ |
| 4 | $5.95025533669726278\ldots \cdot 10^{-4}$ | $+5.95025533669726278\ldots \cdot 10^{-4}$ |

Tabelle 1: $A[j], B[j]$ mit den ersten 18 Dezimalstellen

*Rel. Appr.-Fehler: a[j],b[j] mit 17,16,15,14 dezimalen Stellen*



Die vier Graphen aus obiger Abbildung zeigen den mit $10^{+17}$ multiplizierten relativen Approximationsfehler, wenn die mit Mathematica berechneten Koeffizienten $A_j, B_j$ auf 17, 16, 15 bzw.14 Dezimalstellen der $a_j, b_j$ gerundet werden. Dabei erkennt man z.B., daß sich das Betragsmaximum des relativen Approximationsfehlers fast verachtfacht, wenn die Koeffizienten auf nur 14 Dezimalstellen gerundet werden.

Durch die in der Praxis notwendige Rundung der z.B. mit einer Langzahlarithmetik bestimmten Polynomkoeffizienten $A_j, B_j$ sind die relativen Extremwerte nicht mehr betragsgleich, so daß eine auf die $a_j, b_j$ bezogene Abschätzung des absoluten oder relativen Approximationsfehlers stets erforderlich ist.

## 6.3 Abschätzung des Approximationsfehlers

Nach Vorstellung des grundsätzlichen Lösungswegs im ersten Abschnitt und nach Demonstration des Rundungseinflusses der Polynomkoeffizienten auf den Approximationsfehler werden jetzt Formeln zusammengestellt, mit deren Hilfe garantierte Oberschranken des absoluten bzw. relativen Approximationsfehlers bzgl. der Näherung

$$H(x) \approx \frac{\mathrm{P}_N(x-x_0)}{\mathrm{Q}_M(x-x_0)}, \quad \mathrm{Q}_M(x-x_0) \neq 0, \quad |x-x_0| \leq \eta \tag{9}$$

(mit einem geeigneten XSC-Programm) berechnen werden können. Mit den Approximationspolynomen (die Koeffizienten sind Maschinenzahlen)

$$\mathrm{P}_N(x-x_0) := \sum_{j=0}^{N} a_j \cdot (x-x_0)^j, \quad \mathrm{Q}_M(x-x_0) := \sum_{j=0}^{M} b_j \cdot (x-x_0)^j \tag{10}$$

und der Taylorentwicklung der Hilfsfunktion $H(x)$ um den Punkt $x_0$

$$H(x) = \mathrm{T}_K(x-x_0) + \mathrm{R}_K(x) \tag{11}$$

$$\mathrm{T}_K(x-x_0) := \sum_{j=0}^{K} s_j \cdot (x-x_0)^j$$

$$\mathrm{R}_K(x) := \frac{H^{(K+1)}(\zeta)}{(K+1)!} \cdot (x-x_0)^{K+1}, \quad \zeta = \zeta(x) \text{ zwischen } x \text{ und } x_0 \tag{12}$$

gelten für den absoluten und relativen Approximationsfehler die Abschätzungen

$$|\Delta(x)| \leq \left| \frac{\mathrm{Q}_M(x-x_0) \cdot \mathrm{T}_K(x-x_0) - \mathrm{P}_N(x-x_0)}{\mathrm{Q}_M(x-x_0)} \right| + |\mathrm{R}_K(x)|\,,$$

$$|\varepsilon(x)| \leq \left| \frac{\mathrm{Q}_M(x-x_0) \cdot \mathrm{T}_K(x-x_0) - \mathrm{P}_N(x-x_0)}{H(x) \cdot \mathrm{Q}_M(x-x_0)} \right| + \left| \frac{\mathrm{R}_K(x)}{H(x)} \right|, \quad H(x) \neq 0\,.$$

Da sich in der Praxis die Polynomgrade $N, M$ in der Regel höchstens um 1 unterscheiden werden, ist es keine wirkliche Einschränkung, wenn im folgenden

$M + K \geq N$ vorausgesetzt sein soll. Die Grundidee ist nun, obiges Zählerpolynom zu berechnen, indem man die Differenzen $z_j$ der Polynomkoeffizienten mit einer Intervall-Langzahlarithmetik auswertet

$$\mathrm{Q}_M \cdot \mathrm{T}_K - \mathrm{P}_N \equiv \mathrm{Z}_{M+K}(x - x_0) := \sum_{j=0}^{M+K} z_j \cdot (x - x_0)^j \,.$$

Durch Einsetzen erhält man dann die folgenden Abschätzungen:

$$|\Delta(x)| \leq \left|\frac{\mathrm{Z}_{M+K}(x - x_0)}{\mathrm{Q}_M(x - x_0)}\right| + |\mathrm{R}_K(x)|, \quad |x - x_0| \leq \eta, \tag{13}$$

$$|\varepsilon(x)| \leq \left|\frac{\mathrm{Z}_{M+K}(x - x_0)}{H(x) \cdot \mathrm{Q}_M(x - x_0)}\right| + \left|\frac{\mathrm{R}_K(x)}{H(x)}\right|, \quad H(x) \neq 0 \tag{14}$$

Zur Berechnung der Oberschranken von $|\Delta(x)|, |\varepsilon(x)|$ bzgl. $|x - x_0| \leq \eta$ sind die rechten Seiten von (13),(14) **intervallmäßig** auszuwerten. Wegen der dabei auftretenden Überschätzungen muß der Bereich $|x - x_0| \leq \eta$ in eine jeweils hinreichend große Anzahl von Intervallen unterteilt werden.

**Abschätzung des Restgliedes in (13) bzw. in (14)**

Nach (12) gelten mit

$$u := \frac{H^{(K+1)}([x_0 - \eta, x_0 + \eta])}{(K+1)!}$$

die Aussagen

$$\mathrm{R}_K(x) \in u \cdot [-\eta^{K+1}, +\eta^{K+1}], \quad |\mathrm{R}_K(x)| \leq |u| \cdot \eta^{K+1} =: \tau, \quad \mathrm{R}_K(x) \in [-\tau, +\tau].$$

Das Intervall $u$ wird durch automatische Differentiation mit Hilfe des Moduls `mpitaylor` [4] berechnet, wobei $[x_0 - \eta, x_0 + \eta]$ zur Vermeidung von Überschätzungen in hinreichend viele Teilintervalle $[x]_j$ unterteilt werden kann; in diesem Fall ist dann $|u|$ das Maximum der entsprechenden Teilintervall-Obergrenzen $|u_j|$.

In (14) muß $\mathrm{R}_K(x)$ noch durch $H(x)$ dividiert werden. Da bei der automatischen Differentiation die Funktionswerteinschließung $H(u_j)$ in jedem einzelnen Teilintervalle mitgeliefert wird, dividiert man dazu $|u| \cdot \eta^{K+1}$ noch durch das Minimum der Werte der berechneten $\langle\, |H([x]_j)| \,\rangle$. Falls dieses Minimum verschwindet, wird der Quotient auf `MaxReal` gesetzt, um anzuzeigen, daß eine Oberschranke des relativen Approximationsfehlers nicht berechnet werden kann.

**Abschätzung des ersten Summanden in (13) bzw. in (14)**

In (13) gelten zunächst mit der Abkürzung

$$v := \frac{\mathrm{Z}_{M+K}([x_0 - \eta, x_0 + \eta] - x_0)}{\mathrm{Q}_M([x_0 - \eta, x_0 + \eta] - x_0)}$$

die Abschätzung

$$\left|\frac{\mathrm{Z}_{M+K}(x - x_0)}{\mathrm{Q}_M(x - x_0)}\right| \leq |v|.$$

Falls $|v|$ bei einem zu breiten Intervall $[x_0-\eta, x_0+\eta]$ wegen Überschätzungen bei den Polynomauswertungen von $\mathrm{Z}_{M+K}, \mathrm{Q}_M$ nach dem Intervall-Hornerschema zu groß ausfällt, muß $[x_0-\eta, x_0+\eta]$ wieder in eine hinreichend große Anzahl von Teilintervallen unterteilt werden.

In (14) muß im ersten Summanden rechts zusätzlich noch durch $H(x)$ dividiert werden, was bei komplizierteren Zielfunktionen die Laufzeit erheblich vergrößert. Dies kann man jedoch dadurch vermeiden, daß man $H(x)$ nach (11) durch ihre schon berechnete Taylordarstellung mit Restglied ersetzt:

$$|\varepsilon(x)| \le \left| \frac{\mathrm{Z}_{M+K}(x-x_0)}{[\mathrm{T}_K(x-x_0)+\mathrm{R}_K(x)]\cdot \mathrm{Q}_M(x-x_0)} \right| + \left| \frac{\mathrm{R}_K(x)}{H(x)} \right|, \quad H(x) \neq 0.$$

Da $|\mathrm{R}_K(x)|$ bereits durch $\tau$ abgeschätzt wurde, gilt $\mathrm{R}_K(x) \in [-\tau, +\tau]$, und der erste Summand rechts kann, wie im vorhergehenden Absatz beschrieben, für $x \in [x_0-\eta, x_0+\eta]$ intervallmäßig ausgewertet werden.

# 7 Die Fehlerfunktionen erf$(x)$ und erfc$(x)$

Abbildung 1 zeigt für die Fehlerfunktion

$$\mathrm{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$

und die komplementäre Fehlerfunktion

$$\mathrm{erfc}(x) := 1 - \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2}\, dt$$

die zugehörigen Funktionsgraphen. Für diese Funktionen sollen für reelle Argumente $x$ schnelle Algorithmen mit garantierten Fehlerschranken hergeleitet werden.

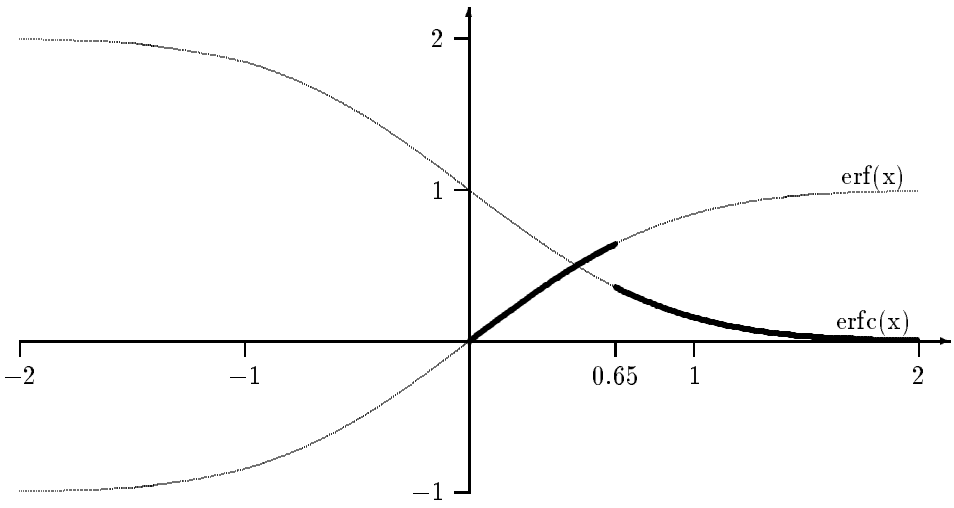



Abbildung 1: erf$(x)$ und erfc$(x)$

In der Literatur wird z.B. in [2, 19, 20, 21, 22, 27, 28] eine Vielzahl von Näherungsfunktionen angegeben. Man findet jedoch entweder nur absolute Fehlerschranken, oder die relativen Fehler werden nur asymptotisch abgeschätzt, so daß konkrete und garantierte Fehlerschranken des relativen Approximationsfehlers nicht zur Verfügung stehen. Werden die Näherungsfunktionen durch Polynome dargestellt, so gibt es zwar Hinweise auf mögliche Auslöschungseffekte, eine Abschätzung des Polynomauswertefehlers wird aber nicht vorgenommen, so daß eine garantierte Fehlerschranke für die Auswertung von $\mathrm{erf}(x)$ bzw. $\mathrm{erfc}(x)$ für ein spezielles Datenformat nicht existiert.

Die gesuchten Algorithmen für die Funktionen $\mathrm{erf}(x)$ und $\mathrm{erfc}(x)$ werden im IEEE-double-Format der Sprache PASCAL-XSC implementiert, wobei angenommen wird, daß die Grundoperationen nur hochgenau (1 Ulp Genauigkeit) ausgeführt werden. Mit den berechneten Fehlerschranken werden entsprechende Intervallfunktionen realisiert.

## 7.1 Grobalgorithmus

| Bereich | Approximation für $\mathrm{erf}(x)$ | | Approximation für $\mathrm{erfc}(x)$ |
|---|---|---|---|
| $B_\circ = 0$ | $0$ | | $1$ |
| $(B_\circ, B_1]$ | $0$ bzw. Warnung | | $1$ |
| $(B_1, B_2]$ | $x \cdot \dfrac{2}{\sqrt{\pi}}$ | $\Longrightarrow$ | $1 - x \cdot \dfrac{2}{\sqrt{\pi}}$ |
| $(B_2, B_3]$ | $x \cdot \dfrac{p_2(x^2) \in P_4}{q_2(x^2) \in P_4}$ | $\Longrightarrow$ | $1 - x\dfrac{p_2(x^2)}{q_2(x^2)}$ |
| $(B_3, B_4]$ | $1 - e^{-x^2}\dfrac{p_3(x)}{q_3(x)}$ | $\Longleftarrow$ | $e^{-x^2}\dfrac{p_3(x) \in P_5}{q_3(x) \in P_6}$ |
| $(B_4, B_5]$ | $1 - e^{-x^2}\dfrac{p_4(x)}{q_4(x)}$ | $\Longleftarrow$ | $e^{-x^2}\dfrac{p_4(x) \in P_5}{q_4(x) \in P_6}$ |
| $(B_5, B_6]$ | $1$ | | $\dfrac{1}{x} \cdot e^{-x^2}\dfrac{p_5(\frac{1}{x^2}) \in P_4}{q_5(\frac{1}{x^2}) \in P_4}$ |
| $> B_6$ | $1$ | | $0$ bzw. Warnung |
| $x < 0$ | $\mathrm{erf}(x) = -\mathrm{erf}(\lvert x \rvert)$ | | $\mathrm{erfc}(x) = 2 - \mathrm{erfc}(\lvert x \rvert)$ |

Die obige Tabelle zeigt übersichtsartig die verschiedenen Approximationen für $\mathrm{erf}(x)$ und $\mathrm{erfc}(x)$ in den einzelnen Teilbereichen. Diese sind durch die folgenden Schranken gegeben:

$$\begin{aligned} B_0 &:= 0,\ B_1 := 1.97193 \cdot 10^{-308},\ B_2 := 10^{-10},\ B_3 := 0.65, \\ B_4 &:= 2.2,\ B_5 := 6,\ B_6 := 26.5432\,. \end{aligned}$$

Pfeile deuten an, wo die eigentliche Grundapproximation leicht modifiziert auch für die jeweils komplementäre Funktion eingesetzt wird. Eine Kennzeichnung $p_n(x) \in P_k$ gibt an, daß das Zählerpolynom $p_n(x)$ im Bereich $(B_n, B_{n+1}]$ verwendet wird und den Grad $k$ besitzt. Entsprechend sind die Angaben für auftretende Nennerpolynome zu interpretieren.

Es ist also nicht notwendig, für $\mathrm{erf}(x)$ und $\mathrm{erfc}(x)$ jeweils einen für alle $x$ kompletten Algorithmus anzugeben! Nur in den fettgedruckten Bereichen der Abb. 1 müssen Näherungsfunktionen für $\mathrm{erf}(x)$ bzw. $\mathrm{erfc}(x)$ angegeben werden:

$$\begin{aligned} &\mathrm{erf}(x) \approx G(x), && x \in [0,\ 0.65] && =: \mathbf{A}, \\ &\mathrm{erfc}(x) \approx g(x), && x \in [0.65,\ +\infty) && =: \mathbf{B}. \end{aligned}$$

Mit den Identitäten

$$\begin{aligned} &\mathrm{erf}(x) \equiv 1 - \mathrm{erfc}(x), && x \in [0.65,\ +\infty) && = \mathbf{B}, \\ &\mathrm{erfc}(x) \equiv 1 - \mathrm{erf}(x), && x \in [0,\ 0.65] && = \mathbf{A}, \\ &\mathrm{erf}(x) \equiv -\mathrm{erf}(-x), && x \in (-\infty, 0] && =: \mathbf{C}, \\ &\mathrm{erfc}(x) \equiv 1 + \mathrm{erf}(-x), && x \in (-\infty, 0] && = \mathbf{C} \end{aligned}$$

lassen sich dann die Funktionen $\mathrm{erf}(x)$ und $\mathrm{erfc}(x)$ mit Hilfe von $G(x)$ und $g(x)$ für alle gewünschten Argumente $x$ auswerten. Bezüglich $\mathrm{erf}(x)$ bzw. $\mathrm{erfc}(x)$ müssen die Intervalle **A** bzw. **B** noch in weitere Teilintervalle unterteilt werden:

$$\begin{aligned} &\mathrm{erf}(x) \text{ im Unterlaufbereich}, && x \in [0,\ 1.97193 \cdot 10^{-308}) && =: \mathbf{A}_0, \\ &\mathrm{erf}(x) \approx G_1(x), && x \in [1.97193 \cdot 10^{-308},\ 10^{-10}] && =: \mathbf{A}_1, \\ &\mathrm{erf}(x) \approx G_2(x), && x \in [10^{-10},\ 0.65] && =: \mathbf{A}_2, \end{aligned}$$

$$\begin{aligned} &\mathrm{erfc}(x) \approx g_1(x), && x \in [0.65,\ 2.2] && =: \mathbf{B}_1, \\ &\mathrm{erfc}(x) \approx g_2(x), && x \in [2.2,\ 6] && =: \mathbf{B}_2, \\ &\mathrm{erfc}(x) \approx g_3(x), && x \in [6,\ 26.5432] && =: \mathbf{B}_3, \\ &\mathrm{erfc}(x) \text{ im Unterlaufbereich}, && x \in [26.5432,\ +\infty) && =: \mathbf{B}_4. \end{aligned}$$

In den Fehlerabschätzungen werden anstelle der im Zahlformat nicht darstellbaren Intervallendpunkte entsprechende Einschließungen dieser Werte verwendet. In den Teilintervallen $\mathbf{A}_0$ und $\mathbf{B}_4$ können die Funktionswerte von $\mathrm{erf}(x)$ und $\mathrm{erfc}(x)$ in den denormalisierten Zahlenbereich des IEEE-double-Formats fallen, so daß die späteren Fehlerabschätzungen, die sich nur auf den normalisierten Zahlenbereich beziehen, in den genannten Intervallen keine Gültigkeit haben. Für Punktargumente aus $\mathbf{A}_0$ und $\mathbf{B}_4$ werden daher entsprechende Fehlermeldungen erzeugt, während zur Intervallauswertung von $\mathrm{erf}(x)$ und $\mathrm{erfc}(x)$ die Funktionswerte auf Null gesetzt werden.

## 7.2 Approximation von erf$(x)$ in A=[0, 0.65]

Der Teilbereich $\mathbf{A} = [0,\, 0.65]$ wird im folgenden weiter in die zwei Teilbereiche $\mathbf{A}_1 = [0,\, 10^{-10}]$ und $\mathbf{A}_2 = [10^{-10},\, 0.65]$ unterteilt. Zunächst wird der Teilbereich $x \in [0,\, 10^{-10}]$ untersucht. Dabei wird von der Reihendarstellung

$$\mathrm{erf}(x) = \frac{2x}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n \cdot x^{2n}}{n!\,(2n+1)} = \frac{2x}{\sqrt{\pi}} \left[ 1 - \frac{x^2}{3} + \frac{x^4}{10} - + \cdots \right]$$

ausgegangen.

Da diese Reihe [2, Formel 7.1.5] für $|x| < 1$ eine Leibniz-Reihe ist, gilt bezüglich der Approximation

$$G_1(x) := \frac{2}{\sqrt{\pi}} \cdot x \approx \mathrm{erf}(x)$$

für den absoluten Fehler $r(x) := \mathrm{erf}(x) - 2x/\sqrt{\pi}$ die Abschätzung

$$|r(x)| \leq \frac{2x^3}{3 \cdot \sqrt{\pi}}\,.$$

Zusammen mit

$$\mathrm{erf}(x) > \frac{2x}{\sqrt{\pi}} \cdot \left( 1 - \frac{x^2}{3} \right)$$

ergibt sich für den relativen Approximationsfehler $\varepsilon_{\mathrm{app}}(x) := r(x)/\mathrm{erf}(x)$ die Abschätzung

$$|\varepsilon_{\mathrm{app}}(x)| \leq \frac{x^2}{3 - x^2} \leq \frac{10^{-20}}{3 - 10^{-20}} < 3.3334 \cdot 10^{-21} =: \varepsilon(\mathrm{app})\,.$$

Um mit Hilfe von (3) eine Fehlerschranke von $\mathrm{erf}(x)$ in $x \in [0,\, 10^{-10}] \cap S(2,53)$ berechnen zu können, benötigt man noch die Fehlerschranke für die Maschinenauswertung von $G_1(x)$ :

$$\begin{aligned} \tilde{G}_1(x) &= \left[ \widetilde{\frac{2}{\sqrt{\pi}}} \right] \boxdot x = G_1(x) \cdot (1+\varepsilon)(1+2\varepsilon), \qquad |\varepsilon| \leq \varepsilon^* = 2^{-53} \\ &= G_1(x) \cdot (1 + \varepsilon_{G1}); \quad |\varepsilon_{G1}| \leq 3.3307 \cdot 10^{-16} =: \varepsilon(G_1)\,. \end{aligned}$$

Bei der obigen Abschätzung wird vorausgesetzt, daß $2/\sqrt{\pi}$ in $S(2,53)$ maximalgenau gespeichert wird. Mit (3) folgt damit

$$\widetilde{\mathrm{erf}}(x) = \mathrm{erf}(x)(1+\varepsilon_f); \qquad |\varepsilon_f| \le 3.3308 \cdot 10^{-16} =: \varepsilon(f)\,.$$

Da die obige Fehlerabschätzung mit der Fehlerschranke $2\varepsilon^* = 2^{-52}$ durchgeführt wird, muß vorausgesetzt werden, daß die Funktionswerte $\tilde{G}_1(x)$ im normalisierten Zahlenbereich liegen, d.h. es muß

$$\tilde{G}_1(x) = G_1(x)\cdot(1+\varepsilon_f) \ge G_1(x)\cdot[1-\varepsilon(f)] = \frac{2x}{\sqrt{\pi}}\cdot[1-\varepsilon(f)] \ge 2^{-1022}$$

gelten. Die letzte Ungleichung ist erfüllt, falls $x \ge 1.97193 \cdot 10^{-308}$. Für das Teilintervall $\mathbf{A}_1 = [1.97193 \cdot 10^{-308},\, 10^{-10}]$ ergibt sich das Ergebnis

$$\boxed{\widetilde{\mathrm{erf}}(x) = \mathrm{erf}(x)(1+\varepsilon_f); \quad |\varepsilon_f| \le 3.3308 \cdot 10^{-16} = \varepsilon(\mathrm{erf},\mathbf{A}_1), \quad x \in \mathbf{A}_1 \cap S(2,53)\,.}$$

Im Fall $x < 1.97193 \cdot 10^{-308}$ wird bei der Berechnung von $\mathrm{erf}(x)$ eine entsprechende Fehlermeldung erzeugt.
Für den zweiten Teilbereich $\mathbf{A}_2 := [10^{-10},\, 0.65]$ wird die für $|x| < +\infty$ gültige Reihenentwicklung [2, Formel 7.1.6]

$$\mathrm{erf}(x) = \frac{2x}{\sqrt{\pi}}\cdot e^{-x^2}\cdot\sum_{n=0}^{\infty} a_n \cdot x^{2n}, \qquad a_n = \frac{2^n}{1\cdot 3\cdot \ldots \cdot (2n+1)};$$

verwendet. Wegen $a_{n+1}/a_n < 1$ erhält man mit

$$\sum_{n=0}^{\infty} a_n \cdot x^{2n} = \sum_{n=0}^{N} a_n \cdot x^{2n} + \sum_{n=N+1}^{\infty} a_n \cdot x^{2n}$$

die Abschätzung

$$\begin{aligned}\sum_{n=N+1}^{\infty} a_n \cdot x^{2n} &= a_{N+1}\cdot x^{2(N+1)}\left[1 + \frac{a_{N+2}}{a_{N+1}}\cdot x^2 + \frac{a_{N+3}}{a_{N+1}}\cdot x^4 + \cdots\right] \\ &< a_{N+1}\cdot x^{2(N+1)}\sum_{n=0}^{\infty} x^{2n} = \frac{a_{N+1}\cdot x^{2(N+1)}}{1-x^2}, \quad |x| < 1\,.\end{aligned}$$

Mit der Hilfsnäherung

$$\mathrm{erf}(x) \approx H(x) := \frac{2x}{\sqrt{\pi}}\cdot e^{-x^2}\sum_{n=0}^{N} a_n \cdot x^{2n}$$

und wegen der durch die Reihenentwicklung gegebenen Ungleichung

$$\mathrm{erf}(x) \ge \frac{2x}{\sqrt{\pi}}\cdot e^{-x^2}, \quad x \ge 0\,,$$

folgt dann für den relativen Approximationsfehler

$$\begin{aligned} \varepsilon_{\mathrm{app},1} &:= \frac{\mathrm{erf}(x) - H(x)}{\mathrm{erf}(x)}; \\ |\varepsilon_{\mathrm{app},1}| &< \frac{2^{N+1} \cdot x^{2(N+1)}}{1 \cdot 3 \cdot \ldots \cdot (2N+3)} \cdot \frac{1}{1-x^2} \leq \varepsilon(\mathrm{app},1), \quad |x| < 1. \end{aligned}$$

Für $N = 14$ gilt damit im Bereich $\mathbf{A}_2 = [10^{-10}, 0.65]$

$$\mathrm{erf}(x) \approx H(x) = \frac{2x}{\sqrt{\pi}} \cdot e^{-x^2} \sum_{n=0}^{14} a_n \cdot x^{2n}, \qquad \varepsilon(\mathrm{app},1) = 7.2149 \cdot 10^{-19}.$$

Die Hilfsnäherungsfunktion $H(x)$ im IEEE–Format könnte durchaus als Maschinenapproximation für $\mathrm{erf}(x)$ benutzt werden, jedoch wäre die Laufzeit wegen der auftretenden Exponentialfunktion und wegen des hohen Polynomgrades $N = 14$ sehr groß! Viel effektiver ist die Verwendung einer zweiten Approximation $G_2(x)$ mit

$$\mathrm{erf}(x) \approx x \cdot \frac{\mathrm{P}_4(x^2)}{\mathrm{Q}_4(x^2)} =: G_2(x), \qquad \mathrm{P}_4(x^2) = \sum_{n=0}^{4} p_n \cdot x^{2n}, \quad \mathrm{Q}_4(x^2) = \sum_{n=0}^{4} q_n \cdot x^{2n},$$

| $n$ | $p_n := \mathrm{nearest}(\,.\,)$ | $q_n := \mathrm{nearest}(\,.\,)$ |
|---|---|---|
| 0 | $1.12837916709551256 \cdot 10^{+0}$ | $1.00000000000000000 \cdot 10^{+0}$ |
| 1 | $1.35894887627277916 \cdot 10^{-1}$ | $4.53767041780002545 \cdot 10^{-1}$ |
| 2 | $4.03259488531795274 \cdot 10^{-2}$ | $8.69936222615385890 \cdot 10^{-2}$ |
| 3 | $1.20339380863079457 \cdot 10^{-3}$ | $8.49717371168693357 \cdot 10^{-3}$ |
| 4 | $6.49254556481904354 \cdot 10^{-5}$ | $3.64915280629351082 \cdot 10^{-4}$ |

Die in obiger Tabelle angegebenen Dezimalwerte wurden mit Hilfe eines Computeralgebrasystems durch rationale Approximation der Hilfsfunktion $H(x)$ bestimmt. Die Koeffizienten $p_n, q_n$ sind die zu den angegebenen Dezimalwerten jeweils nächstgelegenen IEEE–Zahlen. Für die Bestimmung des Approximationsfehlers darf dann von exakt darstellbaren Approximationskoeffizienten ausgegangen werden.

Für $G_2(x)$ ist der relative Approximationsfehler durch

$$\varepsilon_{\mathrm{app},2}(x) := \frac{H(x) - G_2(x)}{H(x)}, \qquad x \in \mathbf{A}_2 = [10^{-10}, 0.65]$$

gegeben. Mit Hilfe der XSC-Programme `AppErr` und `ErrBound` läßt sich eine garantierte Oberschranke $\varepsilon(\mathrm{app},2)$ für $|\varepsilon_{\mathrm{app},2}(x)|$ bzgl. $x \in \mathbf{A}_2$ berechnen. Man erhält

$$H(x) \approx G_2(x) = x \cdot \frac{\mathrm{P}_4(x^2)}{\mathrm{Q}_4(x^2)}; \quad |\varepsilon_{\mathrm{app},2}(x)| \leq 1.3594 \cdot 10^{-17} = \varepsilon(\mathrm{app},2).$$

Mit den jetzt zur Verfügung stehenden Fehlerschranken $\varepsilon(\mathrm{app},1)$, $\varepsilon(\mathrm{app},2)$ ist man nun in der Lage bzgl. der Näherung

$$\mathrm{erf}(x) \approx G_2(x) = x \cdot \frac{\mathrm{P}_4(x^2)}{\mathrm{Q}_4(x^2)}$$

nach Gleichung (2) eine Oberschranke des zugehörigen relativen Approximationsfehlers

$$\varepsilon_{\mathrm{app}}(x) := \frac{\mathrm{erf}(x) - G_2(x)}{\mathrm{erf}(x)}; \qquad |\varepsilon_{\mathrm{app}}(x)| \le \varepsilon(\mathrm{app}), \quad x \in \mathbf{A}_2 = [10^{-10}, 0.65]$$

berechnen zu können. Es ergibt sich

$$\boxed{\mathrm{erf}(x) \approx G_2(x) = x \cdot \frac{\mathrm{P}_4(x^2)}{\mathrm{Q}_4(x^2)}; \qquad |\varepsilon_{\mathrm{app}}(x)| \le 1.4316 \cdot 10^{-17} =: \varepsilon(\mathrm{app})\,.}$$

Um mit Hilfe von Gleichung (1) eine Fehlerschranke von $\mathrm{erf}(x)$ in $x \in \mathbf{A}_2 \cap S(2,53)$ berechnen zu können, benötigen wir noch die relative Fehlerschranke der Maschinenauswertung von $G_2(x)$. Dazu wird die Darstellung

$$\tilde{G}_2(x) = x \boxdot \tilde{\mathrm{P}}_4(x \boxdot x) \boxslash \tilde{\mathrm{Q}}_4(x \boxdot x) = G_2(x)(1 + \varepsilon_{G2}) \tag{15}$$

verwendet. Die gesuchte Fehlerschranke wird wieder automatisch berechnet. Es ergibt sich für das Zählerpolynom

$$\boxed{\tilde{\mathrm{P}}_4(x \boxdot x) = \mathrm{P}_4(x^2)(1 + \varepsilon_{P4}); \qquad |\varepsilon_{P4}| \le 2.6230 \cdot 10^{-16} = \varepsilon(\mathrm{P}_4)\,.}$$

Für das Nennerpolynom findet man

$$\boxed{\tilde{\mathrm{Q}}_4(x \boxdot x) = \mathrm{Q}_4(x^2)(1 + \varepsilon_{Q4}); \qquad |\varepsilon_{Q4}| \le 3.4600 \cdot 10^{-16} = \varepsilon(\mathrm{Q}_4)\,.}$$

Für $\tilde{G}_2(x)$ gilt nach Gleichung (15):

$$\tilde{G}_2(x) = \frac{x \cdot \mathrm{P}_4(x^2)(1 + \varepsilon_{P4})(1 + 2\varepsilon)^2}{\mathrm{Q}_4(x^2)(1 + \varepsilon_{Q4})} = G_2(x)\frac{(1 + \varepsilon_{P4})(1 + 2\varepsilon)^2}{(1 + \varepsilon_{Q4})} = G_2(x)(1 + \varepsilon_{G2})\,.$$

Die Fehlerschranken $\varepsilon(\mathrm{P}_4), \varepsilon(\mathrm{Q}_4)$ und $|\varepsilon| \le \varepsilon^* = 2^{-53}$ liefern schließlich für $|\varepsilon_{G2}|$

$$\boxed{\tilde{G}_2(x) = G_2(x)(1 + \varepsilon_{G2}); \qquad |\varepsilon_{G2}| \le 1.0524 \cdot 10^{-15} = \varepsilon(G_2)\,.}$$

Bzgl. $\widetilde{\mathrm{erf}}(x) = \mathrm{erf}(x)(1 + \varepsilon_f)$ findet man mit Hilfe von Gleichung (3) schließlich eine Abschätzung für $|\varepsilon_f|$ in $\mathrm{A}_2 = [10^{-10}, 0.65]$:

$$\boxed{\widetilde{\mathrm{erf}}(x) = \mathrm{erf}(x)(1+\varepsilon_f); \quad |\varepsilon_f| \le 1.0668 \cdot 10^{-15} = \varepsilon(\mathrm{erf}, \mathbf{A}_2),\ x \in \mathbf{A}_2 \cap S(2,53)\,.}$$

## 7.3 Fehlerschranke für erfc($x$) in A= [0, 0.65]

Mit Hilfe der im letzten Abschnitt bestimmten Maschinenapproximation $\widetilde{\mathrm{erf}}(x)$ und der zugehörigen relativen Fehlerschranke wird nun $\mathrm{erfc}(x) \equiv 1 - \mathrm{erf}(x)$ auf der Maschine wie folgt berechnet:

$$\widetilde{\mathrm{erfc}}(x) := \begin{cases} 1 & : \quad x \in \mathbf{A}_0 = [0,\ 1.97193 \cdot 10^{-308}\,) \\ 1 \boxminus \widetilde{\mathrm{erf}}(x) & : \quad x \in \mathbf{D} := \mathbf{A}_1 \cup \mathbf{A}_2 = [1.97193 \cdot 10^{-308},\ 0.65] \end{cases} .$$

Es wird zunächst das Teilintervall $\mathbf{D} = [1.97193 \cdot 10^{-308},\ 0.65]$ betrachtet. Für die Darstellung

$$\widetilde{\mathrm{erfc}}(x) := 1 \boxminus \widetilde{\mathrm{erf}}(x) = \mathrm{erfc}(x) \cdot (1 + \varepsilon)$$

erhält man für $|\varepsilon|$ die Abschätzung

$$\begin{aligned} |\varepsilon| &\leq 2 \cdot \varepsilon^* + [1 + 2 \cdot \varepsilon^*] \cdot \frac{\max\limits_{x \in \mathbf{D}} \{\mathrm{erf}(x)\} \cdot \varepsilon(\mathrm{erf}, \mathbf{A}_2)}{\min\limits_{x \in \mathbf{D}} \{\mathrm{erfc}(x)\}}; \quad \varepsilon^* = 2^{-53} \\ &= 2 \cdot \varepsilon^* + [1 + 2 \cdot \varepsilon^*] \cdot \frac{\mathrm{erf}(0.65) \cdot \varepsilon(\mathrm{erf}, \mathbf{A}_2)}{1 - \mathrm{erf}(0.65)} . \end{aligned}$$

Zur Abschätzung der rechten Seite benötigt man eine Oberschranke des Funktionswertes erf(0.65). Nach der Definition von $\varepsilon_{\mathrm{app},1}$ (vergl. Seite 20) folgt für $x \in \mathbf{D}$

$$\mathrm{erf}(x) = \frac{H(x)}{1 - \varepsilon_{\mathrm{app},1}} \leq \frac{H(x)}{1 - \varepsilon(\mathrm{app}, 1)} = \frac{2}{\sqrt{\pi}} \cdot \frac{x \cdot e^{-x^2}}{1 - \varepsilon(\mathrm{app}, 1)} \cdot \sum_{n=0}^{14} a_n \cdot x^{2n} .$$

Wertet man hier den Term nach dem letzten Gleichheitszeichen für $x = [0.65,\ 0.65]$ intervallmäßig aus, so erhält man eine Einschließung der gesuchten Oberschranke von erf(0.65). Das nachfolgende XSC-Programm liefert eine garantierte Oberschranke des Wertes des Ausdrucks $\mathrm{erf}(0.65)/(1 - \mathrm{erf}(0.65))$ Die Oberschranke OS hat den numerischen Wert

$$\frac{\mathrm{erf}(0.65)}{\mathrm{erfc}(0.65)} < 1.793525 =: \mathrm{OS} .$$

```
program OS;
{*********************************************************}
{*  Berechnet wird die Einschliessung einer Oberschranke  *}
{*  von   erf(0.65)/( 1-erf(0.65)                         *}
{*********************************************************}
use i_ari;
var a                : array[0..14] of interval;
    k                : integer;
    c, x, x2, eps, t : interval;
begin
  a[0] := 1;
  for k := 1 to 14 do
  a[k] := 2 * a[k-1] / (2*k + 1);
```

```
    c := 1 / SQRT( arctan(intval(1)) );
    { c: Einschliessung von 2/SQRT(Pi) }
    x := 65 / intval(100);
    x2 := x * x;
    eps := intval( (<7.2149e-19),(>7.2149e-19) );
    t := a[14];                { Hornerschema zur }
    for k := 13 downto 0 do    { Berechnung der   }
       t := t * x2 + a[k];     { Summe            }
    t := c * x * exp(-x2) * t / (1 - eps);
    t := t / (1 - t);
    { t: Einschliessung einer Oberschranke }
    { von erf(0.65)/( 1-erf(0.65) )        }
    writeln(sup(t));
end.
```

Für $|\varepsilon|$ erhält man mit der Oberschranke $\mathrm{OS} = 1.793525$ die Abschätzung

$$|\varepsilon| \leq 2 \cdot \varepsilon^* + [1 + 2 \cdot \varepsilon^*] \cdot \mathrm{OS} \cdot \varepsilon(\mathrm{erf}, \mathbf{A}_2) < 2.1354 \cdot 10^{-15} = \varepsilon(\mathrm{erfc}, \mathbf{D}),$$

d. h.

$$\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon); \quad |\varepsilon| \leq 2.1354 \cdot 10^{-15} = \varepsilon(\mathrm{erfc}, \mathbf{D}),\ x \in \mathbf{D} \cap S(2,53).$$

Für den verbleibenden zweiten Teilbereich $\mathbf{A}_0 = [0,\ 1.97193 \cdot 10^{-308}\ )$ findet man für den relativen Approximationsfehler

$$\varepsilon_{\mathrm{app}}(x) := \frac{1 - \mathrm{erfc}(x)}{\mathrm{erfc}(x)} = \frac{1}{\mathrm{erfc}(x)} - 1 = \frac{\mathrm{erf}(x)}{1 - \mathrm{erf}(x)} < \frac{\mathrm{erf}(1.97193 \cdot 10^{-308})}{1 - \mathrm{erf}(1.97193 \cdot 10^{-308})}.$$

Die Reihenentwicklung für $\mathrm{erf}(x)$ von Seite 18 zeigt, daß der relative Approximationsfehler von der Größenordnung $10^{-308}$ ist. Da in diesem Teilbereich als Näherungsfunktion die (exakt darstellbare) Konstante 1 benutzt wird, kommt kein zusätzlicher Auswertefehler hinzu, so daß der relative Fehler bzgl. $\mathrm{erfc}(x)$ in diesem Teilintervall sicher kleiner ist als $\varepsilon(\mathrm{erfc}, \mathbf{D}) = 2.1354 \cdot 10^{-15}$. Damit ist gezeigt, daß

$$\boxed{\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon); \quad |\varepsilon| \leq 2.1354 \cdot 10^{-15} = \varepsilon(\mathrm{erfc}, \mathbf{A}),\ x \in \mathbf{A} \cap S(2,53)}$$

gilt.

## 7.4 Approximation von erfc$(x)$ in $\mathbf{B}_1 \cup \mathbf{B}_2 = [0.65,\ 6]$

In diesem Abschnitt wird nun im Intervall $[0.65,\ 6]$ die Funktion $\mathrm{erfc}(x)$ durch eine Hilfsfunktion $H(x)$ approximiert, welche durch Standardfunktionen realisiert werden kann. Für den zugehörigen Approximationsfehler wird eine garantierte Oberschranke hergeleitet.
Zunächst gilt nach [2, 7.4.11] für $x > 0$ die Integraldarstellung

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \equiv \frac{2x \cdot e^{-x^2}}{\pi} \int_0^\infty \frac{e^{-t^2}}{t^2 + x^2}\, dt, \quad x > 0.$$

Wendet man auf das letzte Integral die Trapezregel der numerischen Integration an, so erhält man [20, S. 114], [22, S. 134]:

$$\operatorname{erfc}(x) = \mathrm{T}(x,h) + \mathrm{P}(x,h), \qquad x > 0,$$

$$\begin{aligned} \mathrm{T}(x,h) &:= \frac{2x\cdot h}{\pi}\cdot e^{-x^2}\left[\frac{1}{2x^2} + \sum_{k=1}^{\infty}\frac{e^{-k^2h^2}}{k^2h^2+x^2}\right], \\ \mathrm{P}(x,h) &:= -\sum_{r=1}^{\infty} G_r, \qquad G_r := \frac{4x\cdot e^{-x^2}}{\pi}\int_0^{\infty}\frac{e^{-t^2}\cdot\cos\,\omega t}{t^2+x^2}\,dt, \quad \omega := \frac{2\pi r}{h}. \end{aligned}$$

Nach [25] läßt sich das Fourier–Integral $G_r$ schreiben als

$$G_r = e^{\omega x}\cdot\operatorname{erfc}(x+\frac{\omega}{2}) + e^{-\omega x}\cdot\operatorname{erfc}(x-\frac{\omega}{2}); \qquad \omega := \frac{2\pi r}{h}\,. \tag{16}$$

Mit der Hilfsapproximationsfunktion

$$H(x) := \frac{2x\cdot h}{\pi}\cdot e^{-x^2}\left[\frac{1}{2x^2} + \sum_{k=1}^{N}\frac{e^{-k^2h^2}}{h^2k^2+x^2}\right] \approx \operatorname{erfc}(x)$$

kann der Betrag des absoluten Approximationsfehlers $\Delta$ abgeschätzt werden durch

$$|\Delta| := |H(x) - \operatorname{erfc}(x)| \le \frac{2xh\cdot e^{-x^2}}{\pi}\cdot\sum_{k=N+1}^{\infty}\frac{e^{-h^2k^2}}{h^2k^2+x^2} + |\mathrm{P}(x,h)| \tag{17}$$

$$\sum_{k=N+1}^{\infty}\frac{e^{-h^2k^2}}{h^2k^2+x^2} < \sum_{k=N+1}^{\infty}\frac{e^{-h^2k^2}}{h^2k^2} < \frac{1}{h^2(N+1)^2}\sum_{k=N+1}^{\infty}e^{-h^2k^2};$$

$$\begin{aligned} \sum_{k=N+1}^{\infty}e^{-h^2k^2} &= \sum_{k=0}^{\infty}e^{-h^2([N+1]+k)^2} = e^{-h^2(N+1)^2}\cdot\sum_{k=0}^{\infty}e^{-h^2k(2[N+1]+k)} \\ &< e^{-h^2(N+1)^2}\cdot\sum_{k=0}^{\infty}e^{-4h^2\cdot k} = \frac{e^{-h^2(N+1)^2}}{1-e^{-4h^2}}\,. \end{aligned}$$

Somit gilt

$$|\Delta| \le \frac{2x\cdot e^{-x^2}}{\pi h\cdot(N+1)^2}\cdot\frac{e^{-h^2(N+1)^2}}{1-e^{-4h^2}} + |\mathrm{P}(x,h)|\,. \tag{18}$$

Zur Abschätzung von $|\mathrm{P}(x,h)|$ verwendet man wegen $\operatorname{erfc}(x) < 2$ zunächst die Ungleichung

$$G_r < e^{\omega x}\cdot\operatorname{erc}(x+\frac{\omega}{2}) + 2\cdot e^{-\omega x} < e^{\omega x}\cdot\operatorname{erfc}(\frac{\omega}{2}) + 2\cdot e^{-\omega x}\,,$$

woraus mit [2, Formel 7.1.13]

$$\operatorname{erfc}\left(\frac{\omega}{2}\right) < \frac{2\cdot e^{-\omega^2/4}}{\sqrt{\pi}\cdot\omega}$$

sowie mit $\omega = 2\pi r/h$ und $r \geq 1$ folgt, daß

$$G_r < \frac{h}{\pi^{3/2}} \cdot e^{-\frac{2\pi r}{h}\left(\frac{\pi}{2h} - x\right)} + 2 \cdot e^{-\frac{2\pi x}{h}} \cdot r, \qquad r = 1, 2, \ldots . \tag{19}$$

Die notwendige Summation über $r$ kann jetzt unter der Voraussetzung $2h \cdot x < \pi$ mit Hilfe der geometrischen Reihe vorgenommen werden. Es ergibt sich

$$|\mathrm{P}(x,h)| < \frac{h}{\pi^{3/2}} \cdot \frac{1}{e^{\frac{2\pi}{h}\left(\frac{\pi}{2h} - x\right)} - 1} + \frac{2}{e^{\frac{2\pi \cdot x}{h}} - 1}; \qquad 2h \cdot x < \pi\,. \tag{20}$$

Zur Berechnung einer **relativen** Fehlerschranke muß jetzt (18) nur noch durch eine Unterschranke von $\mathrm{erfc}(x)$ dividiert werden. Mit der Ungleichung [22, S. 137,(1)]

$$\mathrm{erfc}(x) > \frac{2x}{\sqrt{\pi}} \cdot \frac{e^{-x^2}}{1 + 2x^2}, \qquad x > 0$$

erhält man schließlich zusammen mit (18) und (20) die Ergebnisse:

$$\begin{aligned} \mathrm{erfc}(x) &\approx H(x) = \frac{2xh}{\pi} \cdot e^{-x^2} \left[ \frac{1}{2x^2} + \sum_{k=1}^{N} \frac{e^{-h^2k^2}}{h^2k^2 + x^2} \right]; \\ \mathrm{erfc}(x) &= H(x)(1 + \varepsilon_{\mathrm{app},1}); \qquad x \in [0.65,\, 6]; \end{aligned}$$

$$\begin{aligned} \mathrm{U}_1(x,h,N) &:= \frac{1 + 2x^2}{\sqrt{\pi} \cdot h \cdot (N+1)^2} \cdot \frac{e^{-h^2(N+1)^2}}{1 - e^{-4h^2}}; \\ \mathrm{U}_2(x,h) &:= \frac{h \cdot (1 + 2x^2) \cdot e^{x^2}}{2\pi x \cdot \left[ e^{\frac{2\pi}{h}\left(\frac{\pi}{2h} - x\right)} - 1 \right]}, \qquad 2hx < \pi; \\ \mathrm{U}_3(x,h) &:= \frac{\sqrt{\pi} \cdot (1 + 2x^2) \cdot e^{x^2}}{x} \cdot \frac{1}{e^{\frac{2\pi x}{h}} - 1}; \end{aligned}$$

$$\boxed{\begin{array}{c} |\varepsilon_{\mathrm{app},1}| < \mathrm{U}_1(x,h,N) + \mathrm{U}_2(x,h) + \mathrm{U}_3(x,h) \leq \varepsilon(\mathrm{app},1;x,h,N); \\ x \in [0.65,\, 6]; \qquad 2hx < \pi\,. \end{array}}$$

Abschließend müssen in den Funktionen $\mathrm{U}_i$ die Parameter $x, h, N$ so gewählt werden, daß man bzgl. des IEEE–Formats eine relative Fehlerschranke erhält, die für alle $x \in [0.65,\, 6]$ von der Größenordung $10^{-18}$ ist.

Für $\mathrm{U}_3(x,h)$ sucht man also zu gegebenem $h > 0$ eine Oberschranke des globalen Maximums bzgl. $x \in [0.65,\, 6]$. Zur Berechnung einer **garantierten** Oberschranke kann das XSC-Programm zur eindimensionalen globalen Optimierung aus [9] verwendet werden, in dem die jetzt mit `u3` bezeichnete Funktion wie folgt zu definieren ist:

```
function u3 (x : DerivType) : DerivType;
var h,pi2,sqrtpi : DerivType;
    c1,c2    : interval;
begin
  h := DerivConst( 93/intval(1000) );
  c1 := ARCTAN( intval(1) );              { pi/4     }
  pi2 := DerivConst( 8*c1 );              { 2*pi     }
  sqrtpi := DerivConst( 2*SQRT(c1) );     { sqrt(pi) }
  u3 := -sqrtpi*(1+2*x*x)*exp(x*x)/(x* (exp(pi2*x/h)-1) );
end;
```

Mit $h = 0.093$ erhält man das Ergebnis

$$\mathrm{U}_3(x, 0.093) \le 6.5046 \cdot 10^{-19}, \qquad x \in [0.65,\, 6]\,.$$

Ganz entsprechend läßt sich mit dem Programm zur globalen Optimierung auch eine Oberschranke für die Funktion $\mathrm{U}_2(x,h)$ berechnen. Zur Vermeidung eines Überlaufs wird jedoch vorher noch die Abschätzung

$$\mathrm{U}_2(x,h) := \frac{h\cdot(1+2x^2)\cdot e^{x^2}}{2\pi x\cdot\left[e^{\frac{2\pi}{h}(\frac{\pi}{2h}-x)}-1\right]} < \frac{h\cdot(1+2x^2)\cdot e^{x^2}}{2\pi x\cdot\left[e^{\frac{2\pi}{h}(\frac{\pi}{2h}-8)}-1\right]}, \quad 2hx<\pi$$

durchgeführt. Für die Funktion u2

```
function u2 (x : DerivType) : DerivType;
var h  : DerivType;
    pi : interval;
begin
  h := DerivConst( 93/intval(1000) );
  pi :=4*ARCTAN( intval(1) );               { pi }
  u2 := -h*(1+2*x*x)*exp(x*x)/
        ( 2*pi*x*(exp( 2*pi/h*(pi/(2*h)-8) )-1) )
end;
```

ergibt sich dann mit $h = 0.093$ die Oberschranke

$$\mathrm{U}_2(x, 0.093) \le 1.0877 \cdot 10^{-246}, \qquad x \in [0.65,\, 6]\,.$$

Zur Abschätzung von $\mathrm{U}_1(x,h,N)$ wird die Funktion u1

```
function u1 (x : DerivType) : DerivType;
var h      : DerivType;
    sqrtpi : interval;
    N      : integer;
begin
  h := DerivConst( 93/intval(1000) );
  sqrtpi :=2*SQRT( ARCTAN(intval(1)) );   { sqrt(pi) }
```

```
        N := 70,
        u1 := -(1+2*x*x)*exp(-h*h*(N+1)*(N+1))/
              ( sqrtpi*h*(N+1)*(N+1)*(1-exp(-4*h*h)) )
    end;
```

mit $h = 0.093$ und $N = 70$ herangezogen. Es ergibt sich

$$\mathrm{U}_1(x, 0.093, 70) \le 3.0002 \cdot 10^{-19}, \qquad x \in [0.65,\, 6].$$

Die berechneten globalen Maxima der drei Funktionen $\mathrm{U}_i$ ermöglichen jetzt die Angabe einer garantierten Oberschranke $\varepsilon(\mathrm{app}, 1)$ des relativen Approximationsfehlers für den Datenbereich $x \in [0.65, 6]$ :

$$\begin{aligned} \mathrm{erfc}(x) &\approx H(x) = \frac{2xh}{\pi} \cdot e^{-x^2} \left[ \frac{1}{2x^2} + \sum_{k=1}^{N} \frac{e^{-h^2k^2}}{h^2k^2 + x^2} \right]; \\ \mathrm{erfc}(x) &= H(x)(1 + \varepsilon_{\mathrm{app},1}); \qquad x \in [0.65,\, 6]; \end{aligned}$$

$$|\varepsilon_{\mathrm{app},1}| < 9.5049 \cdot 10^{-19} = \varepsilon(\mathrm{app}, 1), \quad h = 0.093, \quad N = 70\,.$$

Für die tatsächliche Approximation von $\mathrm{erfc}(x)$ auf der Rechenanlage wird der Bereich $[0.65,\, 6]$ wieder in zwei Unterbereiche unterteilt. In jedem dieser Bereiche wird die Hilfsfunktion $H(x)$ dazu herangezogen, eine effiziente Approximation mit garantierter relativer Fehlerschranke zu finden.

### 7.4.1 $\mathrm{erfc}(x)$ in $\mathbf{B}_1 = [0.65,\, 2.2]$

Es wird zunächst der Bereich $\mathbf{B}_1 = [0.65, 2.2]$ betrachtet. Hier wird auf dem Rechner $\mathrm{erfc}(x)$ durch

$$\mathrm{erfc}(x) \approx e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)} =: g_1(x); \qquad \mathrm{P}_5(x) = \sum_{n=0}^{5} p_n \cdot x^n, \quad \mathrm{Q}_6(x) = \sum_{n=0}^{6} q_n \cdot x^n$$

approximiert.

| $n$ | $p_n :=$ nearest( . ) | $q_n :=$ nearest( . ) |
|---|---|---|
| 0 | $9.99999992049799098 \cdot 10^{-1}$ | $1.00000000000000000 \cdot 10^{+0}$ |
| 1 | $1.33154163936765307 \cdot 10^{+0}$ | $2.45992070144245533 \cdot 10^{+0}$ |
| 2 | $8.78115804155881782 \cdot 10^{-1}$ | $2.65383972869775752 \cdot 10^{+0}$ |
| 3 | $3.31899559578213215 \cdot 10^{-1}$ | $1.61876655543871376 \cdot 10^{+0}$ |
| 4 | $7.14193832506776067 \cdot 10^{-2}$ | $5.94651311286481502 \cdot 10^{-1}$ |
| 5 | $7.06940843763253131 \cdot 10^{-3}$ | $1.26579413030177940 \cdot 10^{-1}$ |
| 6 | | $1.25304936549413393 \cdot 10^{-2}$ |

Tabelle 2: Polynomkoeffizienten mit 18 Dezimalstellen

Die angegebenen Dezimalwerte der Approximation an $H(x)$ wurden wieder mit Hilfe eines Computeralgebrasystems bestimmt. Die tatsächlich verwendeten Koeffizienten $p_n, q_n$ sind die zu den angegebenen Dezimalwerten jeweils nächstgelegenen IEEE–Zahlen.

Eine Oberschranke $|\varepsilon_{\mathrm{app},2}(x)|$ für den relativen Approximationsfehler

$$\varepsilon_{\mathrm{app},2}(x) := \frac{H(x) - g_1(x)}{H(x)}, \qquad x \in \mathbf{B}_1 = [0.65,\ 2.2]$$

kann wieder automatisch bestimmt werden. Man findet

$$H(x) \approx g_1(x) := e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)}; \quad |\varepsilon_{\mathrm{app},2}(x)| \leq 1.5772 \cdot 10^{-16} =: \varepsilon(\mathrm{app}, 2)\,.$$

Mit den jetzt zur Verfügung stehenden Fehlerschranken $\varepsilon(\mathrm{app}, 1)$, $\varepsilon(\mathrm{app}, 2)$ ist man in der Lage bzgl. der Näherung

$$g_1(x) := e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)} \approx \mathrm{erfc}(x)$$

nach Gleichung (2) eine Oberschranke des zugehörigen relativen Approximationsfehlers

$$\varepsilon_{\mathrm{app}}(x) := \frac{\mathrm{erfc}(x) - g_1(x)}{\mathrm{erfc}(x)}; \qquad |\varepsilon_{\mathrm{app}}(x)| \leq \varepsilon(\mathrm{app}), \quad x \in \mathbf{B}_1 = [0.65,\ 2.2]$$

zu berechnen. Es ergibt sich

$$\mathrm{erfc}(x) \approx g_1(x) = e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)}; \qquad |\varepsilon_{\mathrm{app}}(x)| \leq 1.5868 \cdot 10^{-16} = \varepsilon(\mathrm{app})\,.$$

Um mit Gleichung (3) eine Fehlerschranke von $\mathrm{erfc}(x)$ in $x \in \mathbf{B}_1 \cap S(2,53)$ berechnen zu können, benötigt man noch die relative Fehlerschranke für die Auswertung von $g_1(x)$ auf der Maschine:

$$\begin{aligned}
\widetilde{g}_1(x) &:= \mathrm{EXPx2}(x) \boxdot \left[\widetilde{\mathrm{P}}_5(x) \boxplus \widetilde{\mathrm{Q}}_6(x)\right] \\
&= e^{-x^2} \cdot (1+\varepsilon_1) \cdot \frac{\widetilde{\mathrm{P}}_5(x)}{\widetilde{\mathrm{Q}}_6(x)} \cdot (1+2\cdot\varepsilon)^2, \quad |\varepsilon_1| \leq \varepsilon(1) \\
&= e^{-x^2} \cdot (1+\varepsilon_1) \cdot \frac{\mathrm{P}_5(x) \cdot (1+\varepsilon_{P5})}{\mathrm{Q}_6(x) \cdot (1+\varepsilon_{Q6})} \cdot (1+2\cdot\varepsilon)^2, \qquad (21) \\
&\qquad |\varepsilon| \leq \varepsilon^* = 2^{-53}, \quad |\varepsilon_{P5}| \leq \varepsilon(\mathrm{P}_5), \quad |\varepsilon_{Q6}| \leq \varepsilon(\mathrm{Q}_6)\,.
\end{aligned}$$

Bedeutet $\mathrm{EXPx2}(x)$ die $e^{-x^2}$-Funktion, so gilt nach Seite 42 bei hochgenauer Arithmetik:

$$\mathrm{EXPx2}(x) = e^{-x^2} \cdot (1+\varepsilon_1), \qquad |\varepsilon_1| \leq 1.0823 \cdot 10^{-15} =: \varepsilon(1)\,.$$

Die Berechnung einer Fehlerschranke $\varepsilon(\mathrm{P}_5)$ für das Zählerpolynom kann wieder automatisch durchgeführt werden, wobei sich

$$\widetilde{\mathrm{P}}_5(x) = \mathrm{P}_5(x)(1+\varepsilon_{P5}); \qquad |\varepsilon_{P5}| \le 1.2027 \cdot 10^{-15} = \varepsilon(\mathrm{P}_5)$$

ergibt. Entsprechend findet man als Fehlerschranke $\varepsilon(\mathrm{Q}_6)$ für das Nennerpolynom

$$\boxed{\widetilde{\mathrm{Q}}_6(x) = \mathrm{Q}_6(x)(1+\varepsilon_{Q6}); \qquad |\varepsilon_{Q6}| \le 1.5838 \cdot 10^{-15} = \varepsilon(\mathrm{Q}_6)\,.}$$

Für die nach Gleichung (21) gültige Darstellung

$$\widetilde{g}_1(x) = g_1(x) \cdot (1+\varepsilon_1) \cdot \frac{1+\varepsilon_{P5}}{1+\varepsilon_{Q6}} \cdot (1+2\cdot\varepsilon)^2 = g_1(x) \cdot (1+\varepsilon_{g1})$$

ergibt sich

$$\widetilde{g}_1(x) = g_1(x)(1+\varepsilon_{g1}); \qquad |\varepsilon_{g1}| \le 4.3129 \cdot 10^{-15} = \varepsilon(g_1).$$

Bezüglich $\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon_f)$ findet man mit Hilfe von Gleichung (3) und den Fehlerschranken $\varepsilon(\mathrm{app}) = 1.5868 \cdot 10^{-16}$, $\varepsilon(g_1) = 4.3129 \cdot 10^{-15}$ für $|\varepsilon_f|$ die endgültige Abschätzung

$$\boxed{\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon_f); \quad |\varepsilon_f| \le 4.4716 \cdot 10^{-15} = \varepsilon(\mathrm{erfc},\mathbf{B}_1), \;\; x \in \mathbf{B}_1 \cap S(2,53)\,.}$$

### 7.4.2 $\mathrm{erfc}(x)$ in $\mathbf{B}_2 = [2.2,\, 6]$

Im Bereich $\mathbf{B}_2 = [2.2,\, 6]$ wird $\mathrm{erfc}(x)$ auf dem Rechner durch

$$\mathrm{erfc}(x) \approx e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)} =: g_2(x); \qquad \mathrm{P}_5(x) = \sum_{n=0}^{5} p_n \cdot x^n, \quad \mathrm{Q}_6(x) = \sum_{n=0}^{6} q_n \cdot x^n$$

approximiert.

| $n$ | $p_n :=$ nearest( . ) | $q_n :=$ nearest( . ) |
|---|---|---|
| 0 | $9.99921140009714409 \cdot 10^{-1}$ | $1.00000000000000000 \cdot 10^{+0}$ |
| 1 | $1.62356584489366647 \cdot 10^{+0}$ | $2.75143870676376208 \cdot 10^{+0}$ |
| 2 | $1.26739901455873222 \cdot 10^{+0}$ | $3.37367334657284535 \cdot 10^{+0}$ |
| 3 | $5.81528574177741135 \cdot 10^{-1}$ | $2.38574194785344389 \cdot 10^{+0}$ |
| 4 | $1.57289620742838702 \cdot 10^{-1}$ | $1.05074004614827206 \cdot 10^{+0}$ |
| 5 | $2.25716982919217555 \cdot 10^{-2}$ | $2.78788439273628983 \cdot 10^{-1}$ |
| 6 | | $4.00072964526861362 \cdot 10^{-2}$ |

Tabelle 3: Polynomkoeffizienten mit 18 Dezimalstellen

Die Dezimalwerte der Tabelle 3 wurden durch Approximation von $H(x)$ mit einem Computeralgebrasystem ermittelt. Die tatsächlich verwendeten Koeffizienten $p_n, q_n$ sind die zu den angegebenen Dezimalwerten jeweils nächstgelegenen IEEE–Zahlen.

Für den Approximationsfehler

$$\varepsilon_{\mathrm{app},2}(x) := \frac{H(x) - g_2(x)}{H(x)}, \qquad x \in \mathbf{B}_2 = [2.2,\, 6]$$

läßt sich wieder automatisch die Oberschranke $|\varepsilon_{\mathrm{app},2}(x)|$ bzgl. $x \in \mathrm{B}_2$ berechnen. Man findet

$$H(x) \approx g_2(x) := e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)}; \quad |\varepsilon_{\mathrm{app},2}(x)| \le 1.5282 \cdot 10^{-16} =: \varepsilon(\mathrm{app}, 2)\,.$$

Mit den jetzt zur Verfügung stehenden Fehlerschranken $\varepsilon(\mathrm{app}, 1)$, $\varepsilon(\mathrm{app}, 2)$ ist man in der Lage, für die Näherung

$$\mathrm{erfc}(x) \approx g_2(x) = e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)}$$

nach Gleichung (2) eine Oberschranke des zugehörigen relativen Approximationsfehlers

$$\varepsilon_{\mathrm{app}}(x) := \frac{\mathrm{erfc}(x) - g_2(x)}{\mathrm{erfc}(x)}; \qquad |\varepsilon_{\mathrm{app}}(x)| \le \varepsilon(\mathrm{app}), \quad x \in \mathbf{B}_2 = [2.2,\, 6]$$

zu berechnen:

$$\mathrm{erfc}(x) \approx g_2(x) = e^{-x^2} \cdot \frac{\mathrm{P}_5(x)}{\mathrm{Q}_6(x)}; \qquad |\varepsilon_{\mathrm{app}}(x)| \le 1.5378 \cdot 10^{-16} = \varepsilon(\mathrm{app})\,.$$

Bevor mit Gleichung (3) eine Fehlerschranke von $\mathrm{erfc}(x)$ in $x \in \mathbf{B}_2 \cap S(2,53)$ berechnet werden kann, muß eine relative Fehlerschranke für die Maschinenauswertung von $g_2(x)$ bestimmt werden, wobei davon ausgegangen werden soll, daß der Faktor $e^{-x^2}$ mit Hilfe der Funktion EXPx2(...) ausgewertet wird. Man findet

$$
\begin{aligned}
\tilde{g}_2(x) &:= \mathrm{EXPx2}(x) \boxdot \left[\tilde{\mathrm{P}}_5(x) \boxplus \tilde{\mathrm{Q}}_6(x)\right] \\
&= e^{-x^2} \cdot (1+\varepsilon_1) \cdot \frac{\tilde{\mathrm{P}}_5(x)}{\tilde{\mathrm{Q}}_6(x)} \cdot (1+2\cdot\varepsilon)^2, \quad |\varepsilon| \le \varepsilon^* = 2^{-53} \\
&= e^{-x^2} \cdot (1+\varepsilon_1) \cdot \frac{\mathrm{P}_5(x)\cdot(1+\varepsilon_{P5})}{\mathrm{Q}_6(x)\cdot(1+\varepsilon_{Q6})} \cdot (1+2\cdot\varepsilon)^2, \qquad (22) \\
&\qquad\qquad |\varepsilon_{P5}| \le \varepsilon(\mathrm{P}_5), \quad |\varepsilon_{Q6}| \le \varepsilon(\mathrm{Q}_6)\,,
\end{aligned}
$$

wobei $\varepsilon_1$ den durch die Exponentialfunktion eingeschleppten Fehler bedeutet (vergl. Seite 42).

In nun schon gewohnter Weise ergeben sich als Fehlerschranke für das Zählerpolynom $\varepsilon(\mathrm{P}_5)$

$$\tilde{\mathrm{P}}_5(x) = \mathrm{P}_5(x)(1+\varepsilon_{P5}); \qquad |\varepsilon_{P5}| \le 1.8701\cdot 10^{-15} = \varepsilon(\mathrm{P}_5)$$

und als Fehlerschranke für das Nennerpolynom $\varepsilon(\mathrm{Q}_6)$

$$\tilde{\mathrm{Q}}_6(x) = \mathrm{Q}_6(x)(1+\varepsilon_{Q6}); \qquad |\varepsilon_{Q6}| \le 2.3036\cdot 10^{-15} = \varepsilon(\mathrm{Q}_6)\,.$$

Die Gleichung (22) liefert die Darstellung

$$\tilde{g}_2(x) = g_2(x)\cdot(1+\varepsilon_1)\cdot\frac{1+\varepsilon_{P5}}{1+\varepsilon_{Q6}}\cdot(1+2\varepsilon)^2 = g_2(x)\cdot(1+\varepsilon_{g2}),$$

für welche

$$\tilde{g}_2(x) = g_2(x)(1+\varepsilon_{g2}); \qquad |\varepsilon_{g2}| \le 5.7002\cdot 10^{-15} = \varepsilon(g_2)$$

gezeigt werden kann. Bezüglich $\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon_f)$ findet man mit Hilfe von Gleichung (3) und den Fehlerschranken

$$\varepsilon(\mathrm{app}) = 1.5378\cdot 10^{-16}, \qquad \varepsilon(g_2) = 5.7002\cdot 10^{-15}$$

für $|\varepsilon_f|$ schließlich die Abschätzung

$$\boxed{\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon_f); \quad |\varepsilon_f| \le 5.8540\cdot 10^{-15} = \varepsilon(\mathrm{erfc},\mathbf{B}_2), \quad x \in \mathbf{B}_2 \cap S(2,53)\,.}$$

## 7.5 Fehlerschranke für erf($x$) in $\mathbf{B}_1 \cup \mathbf{B}_2 = [0.65,\ 6]$

Es wird zunächst der Teilbereich $[0.65,\ 2.2]$ betrachtet. Wegen $\mathrm{erf}(x) \equiv 1 - \mathrm{erfc}(x)$ gilt für die Auswertung auf dem Rechner

$$\widetilde{\mathrm{erf}}(x) := 1 \boxminus \widetilde{\mathrm{erfc}}(x) = \mathrm{erf}(x) \cdot (1 + \varepsilon_0)\,.$$

Damit erhält man

$$\begin{aligned} |\varepsilon_0| &\leq 2 \cdot \varepsilon^* + [1 + 2 \cdot \varepsilon^*] \cdot \frac{\max\limits_{x \in \mathbf{B}_1}\{\mathrm{erfc}(x)\} \cdot \varepsilon(\mathrm{erfc}, \mathbf{B}_1)}{\min\limits_{x \in \mathbf{B}_1}\{\mathrm{erf}(x)\}}; \quad \varepsilon^* = 2^{-53} \\ &= 2 \cdot \varepsilon^* + [1 + 2 \cdot \varepsilon^*] \cdot \frac{[1 - \mathrm{erf}(0.65)] \cdot \varepsilon(\mathrm{erfc}, \mathbf{B}_1)}{\mathrm{erf}(0.65)}. \end{aligned}$$

Zur weiteren Abschätzung der rechten Seite benötigen man zunächst eine Unterschranke von erf(0.65). Nach Definition von $\varepsilon_{\mathrm{app},1}$ (vergl. S. 20) folgt für $x \in \mathbf{A} = [0,\ 0.65]$ :

$$\mathrm{erf}(x) = \frac{H(x)}{1 - \varepsilon_{\mathrm{app},1}} \geq \frac{H(x)}{1 + \varepsilon(\mathrm{app}, 1)} = \frac{2}{\sqrt{\pi}} \cdot \frac{x \cdot e^{-x^2}}{1 + \varepsilon(\mathrm{app}, 1)} \cdot \sum_{n=0}^{14} a_n \cdot x^{2n}.$$

Wertet man hier den Term nach dem letzten Gleichheitszeichen für $x = [0.65,\ 0.65]$ intervallmäßig aus, so erhält man eine Einschließung der gesuchten Unterschranke von erf(0.65). Das folgende Programm verwendet diese Unterschranke, um eine garantierte Oberschranke für den Ausdruck $[1 - \mathrm{erf}(0.65)]/\mathrm{erf}(0.65)$ zu berechnen.

```
program upper_b2;
{*****************************************************}
{*  Berechnet die Einschliessung einer Oberschranke  *}
{*  von   [1-erf(0.65)]/erf(0.65)                    *}
{*****************************************************}
use i_ari;
var a            : array[0..14] of interval;
    k            : integer;
    c,x,x2,eps,t : interval;
begin
  a[0] := 1;
  for k := 1 to 14 do
     a[k] := 2 * a[k-1] / (2*k + 1);
  c := 1 / SQRT( arctan(intval(1)) );
  { c: Einschliessung von 2/SQRT(Pi) }
  x := 65 / intval(100);  x2 := x * x;
  eps := intval( (<7.2149e-19),(>7.2149e-19) );
  t := a[14];             { Hornerschema zur }
  for k := 13 downto 0 do { Berechnung der   }
      t := t * x2 + a[k]; { Summe            }
```

```
  t := c * x * exp(-x2) * t / (1 + eps);
  t := 1 / t - 1;              writeln(t);
  { t: Einschliessung einer Oberschranke }
  { von [1-erf(0.65)] / erf(0.65) )      }
end.
```

Man findet

$$\frac{1-\mathrm{erf}(0.65)}{\mathrm{erf}(0.65)} = \frac{\mathrm{erfc}(0.65)}{\mathrm{erf}(0.65)} < 0.5575613 =: \mathrm{OS}\,.$$

Für $|\varepsilon_0|$ erhält man mit der Oberschranke $\mathrm{OS} = 0.5575613$ die Abschätzung

$$|\varepsilon_0| \le 2\cdot\varepsilon^* + [1+2\cdot\varepsilon^*]\cdot \mathrm{OS}\cdot\varepsilon(\mathrm{erfc},\mathbf{B}_1) < 2.7153\cdot 10^{-15} = \varepsilon(\mathrm{erf},\mathbf{B}_1)\,,$$

also

$$\boxed{\widetilde{\mathrm{erf}}(x) = \mathrm{erf}(x)(1+\varepsilon);\quad |\varepsilon| \le 2.7153\cdot 10^{-15} = \varepsilon(\mathrm{erf},\mathbf{B}_1),\ x\in\mathbf{B}_1\cap S(2,53)\,.}$$

Jetzt wird der zweite Teilbereich $\mathbf{B}_2 = [2.2,\,6]$ betrachtet. Wegen $\mathrm{erf}(x) \equiv 1-\mathrm{erfc}(x)$ gilt für die Auswertung auf dem Rechner

$$\widetilde{\mathrm{erf}}(x) := 1 \boxminus \widetilde{\mathrm{erfc}}(x) = \mathrm{erf}(x)\cdot(1+\varepsilon_0)\,,$$

womit man für den relativen Fehler $|\varepsilon_0|$ bzgl. der Addition fehlerbehafteter Größen die Abschätzung

$$\begin{aligned} |\varepsilon_0| &\le 2\cdot\varepsilon^* + [1+2\cdot\varepsilon^*]\cdot\frac{\max\limits_{x\in\mathbf{B}_2}\{\mathrm{erfc}(x)\}\cdot\varepsilon(\mathrm{erfc},\mathbf{B}_2)}{\min\limits_{x\in\mathbf{B}_2}\{1-\mathrm{erfc}(x)\}};\quad \varepsilon^* = 2^{-53} \\ &= 2\cdot\varepsilon^* + [1+2\cdot\varepsilon^*]\cdot\frac{\mathrm{erfc}(2.2)\cdot\varepsilon(\mathrm{erfc},\mathbf{B}_2)}{1-\mathrm{erfc}(2.2)} \end{aligned}$$

erhält. Zur Abschätzung der rechten Seite benötigen man eine Oberschranke des Funktionswertes erfc(2.2). Nach Definition von $\varepsilon_{\mathrm{app},1}$ (vergl. S. 27) folgt für $x \in \mathbf{B}_2 = [2.2,\,6]$ :

$$\mathrm{erfc}(x) = \frac{H(x)}{1-\varepsilon_{\mathrm{app},1}} \le \frac{H(x)}{1-\varepsilon(\mathrm{app},1)} = \frac{2xh\cdot e^{-x^2}}{\pi\cdot[1-\varepsilon(\mathrm{app},1)]}\left[\frac{1}{2x^2} + \sum_{k=1}^{70}\frac{e^{-h^2k^2}}{h^2k^2+x^2}\right].$$

Wertet man hier den Term nach dem letzten Gleichheitszeichen für $x = [2.2,\,2.2]$ intervallmäßig aus, so erhält man eine garantierte Oberschranke von erfc(2.2). Mit diesen Überlegungen erkennt man, daß das folgende XSC-Programm eine garantierte Oberschranke für $\mathrm{erfc}(2.2)]/[1-\mathrm{erfc}(2.2)]$ liefert.

```
program upper_b3;
{*************************************************}
{*  Berechnet die Einschliessung einer garantierten  *}
```

```
{*                  Oberschranke von                    *}
{*              erfc(2.2) / [1-erfc(2.2)]               *}
{*******************************************************}
use i_ari;
var h,h2,pi,x,x2,eps,a,b,s,t : interval;
    k                        : integer;
begin
  h   := 93 / intval(1000);        h2 := h * h;
  pi  := 4 * arctan( intval(1) );
  x   := 11 / intval(5);
  x2 := x * x;
  eps := intval( (<9.5049e-19),(>9.5049e-19) );
  a   := 2 * x * h * exp(-x2) / (pi * (1-eps) );
  b   := 1 / (2 * x2);    s := 0,
  For k := 1 to 70 do
  s := s + exp(-h2 * k * k) / (h2 * k * k + x2);
  t   := a * (b + s);
  t   := t / (1 - t);
  { t :     Einschliessung einer garantierten  }
  { Oberschranke von  erfc(2.2)/[1-erfc(2.2)]  }
  writeln(t);
end.
```

Man erhält

$$\frac{\mathrm{erfc}(2.2)}{1-\mathrm{erfc}(2.2)} < 1.866323 \cdot 10^{-3} = \mathrm{OS}\,.$$

Für $|\varepsilon_0|$ ergibt sich mit der Oberschranke $\mathrm{OS} = 1.866323 \cdot 10^{-3}$ die Abschätzung

$$|\varepsilon_0| \leq 2 \cdot \varepsilon^* + [1 + 2 \cdot \varepsilon^*] \cdot \mathrm{OS} \cdot \varepsilon(\mathrm{erfc}, \mathbf{B}_2) < 2.3298 \cdot 10^{-16} = \varepsilon(\mathrm{erf}, \mathbf{B}_2)\,,$$

und damit

$$\boxed{\widetilde{\mathrm{erf}}(x) = \mathrm{erf}(x)(1+\varepsilon); \quad |\varepsilon| \leq 2.3298 \cdot 10^{-16} = \varepsilon(\mathrm{erf}, \mathbf{B}_2),\ x \in \mathbf{B}_2 \cap S(2,53)\,.}$$

## 7.6 Approximation von erfc$(x)$ in $\mathbf{B}_3 = [6,\ 26.5432]$

Zunächst werden die Funktionsargumente $x$ nach oben durch 26.5432 beschränkt, um sicherzustellen, daß die Funktionswerte erfc$(x)$ im **normalisierten** Zahlenbereich des IEEE double-Formats liegen:

$$x \in \mathbf{B}_3 = [6,\ 26.5432] \quad \Longrightarrow \quad \mathrm{erfc}(x) > 2^{-1022} = 2.2250738\ldots \cdot 10^{-308}$$

Für $x \geq 6$ wird erfc$(x)$ durch ihre Asymptotik approximiert [2, Formeln 7.1.23, 7.1.24]

$$\mathrm{erfc}(x) = \frac{e^{-x^2}}{\sqrt{\pi} \cdot x}\left[1 - \frac{1}{(2x^2)^1} + \frac{1 \cdot 3}{(2x^2)^2} - + \ldots + (-1)^N \cdot \frac{1 \cdot 3 \cdots (2N-1)}{(2x^2)^N} + r\right]$$

$$|r| \equiv |r(x,N)| \leq \frac{1\cdot 3\cdot 5\cdots(2N+1)}{(2x^2)^{N+1}}; \quad N = 1,2,3,\ldots$$

Für $N = 35$ erhält man durch Intervallrechnung

$$|r(x,35)| \leq |r(6,35)| = \frac{1\cdot 3\cdot 5\cdots 71}{72^{36}} < 3.276507\cdot 10^{-16}\,.$$

Der relative Approximationsfehler ist dann durch

$$|\varepsilon_{\mathrm{app},1}(x)| := \frac{r(6,35)}{\mathrm{erfc}(x)} < \frac{e^{-x^2}}{\sqrt{\pi}\cdot x}\cdot\frac{3.276507\cdot 10^{-16}}{\mathrm{erfc}(x)}$$

gegeben. Die Funktion $\mathrm{erfc}(x)$ besitzt nach [20, S. 201] für $x > 0$ die Unterschranke

$$\frac{2}{\sqrt{\pi}}\cdot\frac{x\cdot e^{-x^2}}{2x^2+1} < \mathrm{erfc}(x), \qquad \text{woraus}$$

$$\begin{aligned} |\varepsilon_{\mathrm{app},1}(x)| &< \left[1+\frac{1}{2x^2}\right]\cdot 3.276507\cdot 10^{-16} \leq \left[1+\frac{1}{72}\right]\cdot 3.276507\cdot 10^{-16} \\ &< 3.322015\cdot 10^{-16} = \varepsilon(\mathrm{app},1); \qquad x \geq 6, \quad N = 35 \end{aligned}$$

folgt.

$$\begin{aligned} \mathrm{erfc}(x) &\approx H(x) := \frac{e^{-x^2}}{\sqrt{\pi}\cdot x}\left[1 - \frac{1}{(2x^2)^1} + \frac{1\cdot 3}{(2x^2)^2} - + \ \ldots\ - \frac{1\cdot 3\cdots 69}{(2x^2)^{35}}\right] \\ \mathrm{erfc}(x) &= H(x)\cdot[1+\varepsilon_{\mathrm{app},1}(x)] \end{aligned}$$

$$|\varepsilon_{\mathrm{app},1}(x)| < 3.322015\cdot 10^{-16} = \varepsilon(\mathrm{app},1), \qquad x \geq 6.$$

Da die numerische Auswertung von $H(x)$ wegen $N = 35$ aus Laufzeitgründen viel zu aufwendig wäre, wird $\mathrm{erfc}(x)$ auf dem Rechner durch

$$\mathrm{erfc}(x) \approx \frac{e^{-x^2}}{x}\cdot\frac{\mathrm{P}_4(\frac{1}{x^2})}{\mathrm{Q}_4(\frac{1}{x^2})} =: g_3(x); \quad \mathrm{P}_4(\frac{1}{x^2}) = \sum_{n=0}^{4} p_n\cdot x^{-2n}, \quad \mathrm{Q}_4(\frac{1}{x^2}) = \sum_{n=0}^{4} q_n\cdot x^{-2n}$$

approximiert.

| $n$ | $p_n :=$ nearest( . ) | $q_n :=$ nearest( . ) |
|---|---|---|
| 0 | $5.64189583547756078\cdot 10^{-1}$ | $1.00000000000000000\cdot 10^{+0}$ |
| 1 | $8.80253746105525775\cdot 10^{+0}$ | $1.61020914205869003\cdot 10^{+1}$ |
| 2 | $3.84683103716117320\cdot 10^{+1}$ | $7.54843505665954743\cdot 10^{+1}$ |
| 3 | $4.77209965874436377\cdot 10^{+1}$ | $1.12123870801026015\cdot 10^{+2}$ |
| 4 | $8.08040729052301677\cdot 10^{+0}$ | $3.73997570145040850\cdot 10^{+1}$ |

Tabelle 4: Dezimalnäherungen für die IEEE-Koeffizienten $p_n, q_n$

Die Dezimalwerte aus obiger Tabelle wurden als rationale Approximation an $H(x)$ bestimmt. Die Koeffizienten $p_n, q_n$ sind die zu den angegebenen Dezimalwerten jeweils nächstgelegenen IEEE–Zahlen. Für die Näherung ist der relative Approximationsfehler gegeben als

$$\varepsilon_{\mathrm{app},2}(x) := \frac{H(x) - g_3(x)}{H(x)}, \qquad x \in [6,\, 27]\,.$$

Mit $u := 1/x^2$ und

$$\mathrm{A}(u) := \frac{1}{\sqrt{\pi}} \cdot \left[1 - \frac{1 \cdot u}{2^1} + \frac{1 \cdot 3 \cdot u^2}{2^2} - + \ \ldots \ - \frac{1 \cdot 3 \cdot \ldots \cdot 69 \cdot u^{35}}{2^{35}}\right] \qquad \text{gilt}$$

$$\varepsilon_{\mathrm{app},2}(x) \equiv \varepsilon_2(u) := \frac{\mathrm{A}(u) - \mathrm{P}_4(u)/\mathrm{Q}_4(u)}{\mathrm{A}(u)}, \qquad u \in [27^{-2}, 6^{-2}]\,.$$

Mit Hilfe des Moduls `mpitaylor` kann die PASCAL-XSC Funktion $H(x)$ durch den einfacheren Ausdruck $\mathrm{A}(u)$ realisiert werden:

```
VAR c : mpinterval;              { c = 1 /sqrt(Pi) }

FUNCTION H(u: mpi_taylor): mpi_taylor[0..ub(u)];
var k   : integer;
    s,a : mpi_taylor[0..ub(u)];
BEGIN
    u:= u / 2;
    s[0]:= 1;
    For k:= 1 to ub(u) do s[k]:= 0,
    a:= s;
    For k:= 1 to 35 do
    begin
      a:= -a * (2*k-1) * u;
      s:= s + a;
    end;
    H:= s * c;
END;  { H }
. . .

BEGIN
  setprec(8); { Definiert ca. 8*8 genaue Stellen }
  c := arctan(_mpinterval(1.0));
  c := 1 / ( 2 * sqrt(c) );      { c = 1/sqrt(Pi) }
END.
```

Nun läßt sich wieder automatisch eine garantierte Oberschranke für $|\varepsilon_{\mathrm{app},2}(x)|$ bzgl. $x \in \mathbf{B}_3$ berechnen.

$$H(x) \approx g_3(x) := \frac{e^{-x^2}}{x} \cdot \frac{\mathrm{P}_4(x^{-2})}{\mathrm{Q}_4(x^{-2})}; \quad |\varepsilon_{\mathrm{app},2}(x)| \leq 9.0000 \cdot 10^{-17} = \varepsilon(\mathrm{app}, 2)\,.$$

Mit den jetzt zur Verfügung stehenden Fehlerschranken $\varepsilon(\mathrm{app},1)$, $\varepsilon(\mathrm{app},2)$ ist man in der Lage bzgl. der Näherung

$$\mathrm{erfc}(x) \approx g_3(x) := \frac{e^{-x^2}}{x} \cdot \frac{\mathrm{P}_4(x^{-2})}{\mathrm{Q}_4(x^{-2})}$$

nach Gleichung (2) eine Oberschranke des zugehörigen relativen Approximationsfehlers

$$\varepsilon_{\mathrm{app}}(x) := \frac{\mathrm{erfc}(x) - g_3(x)}{\mathrm{erfc}(x)}; \qquad |\varepsilon_{\mathrm{app}}(x)| \le \varepsilon(\mathrm{app}), \quad x \in [6,\,27]$$

zu berechnen. Man findet

$$\mathrm{erfc}(x) \approx g_3(x) := \frac{e^{-x^2}}{x} \cdot \frac{\mathrm{P}_4(x^{-2})}{\mathrm{Q}_4(x^{-2})}; \qquad |\varepsilon_{\mathrm{app}}(x)| \le 4.2221 \cdot 10^{-16} = \varepsilon(\mathrm{app}).$$

Um nun für $\mathbf{B}_3 = [6,\,26.5432]$ mit Gleichung (3) eine Fehlerschranke von $\mathrm{erfc}(x)$ in $x \in \mathbf{B}_3 \cap S(2,53)$ berechnen zu können, benötigt man noch die relative Fehlerschranke von $g_3(x)$, wobei davon ausgegangen werden soll, daß der Faktor $e^{-x^2}$ mit Hilfe der Funktion EXPx2(...) berechnet wird.

$$\begin{aligned} \widetilde{g}_3(x) &:= \left[\mathrm{EXP\_X2}(x) \boxdot \widetilde{\mathrm{P}}_4(x \boxdot x)\right] \boxslash \left[x \boxdot \widetilde{\mathrm{Q}}_4(x \boxdot x)\right] \\ (23) \qquad &= \frac{e^{-x^2} \cdot (1+\varepsilon_1) \cdot \mathrm{P}_4(x^2) \cdot (1+\varepsilon_{P4}) \cdot (1+2\cdot\varepsilon)^2}{x \cdot \mathrm{Q}_4(x^2) \cdot (1+\varepsilon_{Q4}) \cdot (1+\varepsilon_k)}, \\ &\qquad |\varepsilon_1| \le \varepsilon(1), \quad |\varepsilon| \le \varepsilon^* = 2^{-53}, \quad |\varepsilon_{P4}| \le \varepsilon(\mathrm{P}_4), \quad |\varepsilon_{Q4}| \le \varepsilon(\mathrm{Q}_4). \end{aligned}$$

Als Fehlerschranke $\varepsilon(\mathrm{P}_4)$ für die maschinelle Zählerpolynomauswerung ergibt sich

$$\widetilde{\mathrm{P}}_4(x^{-2}) = \mathrm{P}_4(x^{-2})(1+\varepsilon_{P4}); \qquad |\varepsilon_{P4}| \le 6.2806 \cdot 10^{-16} = \varepsilon(\mathrm{P}_4)$$

und als Fehlerschranke $\varepsilon(\mathrm{Q}_4)$ für die maschinelle Auswertung des Nennerpolynoms findet man

$$\boxed{\widetilde{\mathrm{Q}}_4(x^{-2}) = \mathrm{Q}_4(x^{-2})(1+\varepsilon_{Q4}); \qquad |\varepsilon_{Q4}| \le 6.4250 \cdot 10^{-16} = \varepsilon(\mathrm{Q}_4)\,.}$$

Die Gleichung (23) liefert die Darstellung

$$\widetilde{g}_3(x) = g_3(x) \cdot \frac{(1+\varepsilon_1)(1+\varepsilon_{P4})(1+2\cdot\varepsilon)^2}{(1+\varepsilon_{Q4})(1+2\cdot\varepsilon)} = g_3(x) \cdot (1+\varepsilon_{g3}),$$

für welche man

$$\widetilde{g}_3(x) = g_3(x)(1+\varepsilon_{g3}); \qquad |\varepsilon_{g3}| \le 3.0190 \cdot 10^{-15} = \varepsilon(g_3)$$

zeigen kann. Bezüglich $\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon_f)$ findet man mit Hilfe von Gleichung (3) und den Fehlerschranken $\varepsilon(\mathrm{app}) = 4.2221 \cdot 10^{-16}$, $\varepsilon(g_3) = 3.0910 \cdot 10^{-15}$ für $|\varepsilon_f|$ die Abschätzung

$$\boxed{\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon_f); \quad |\varepsilon_f| \le 3.4413 \cdot 10^{-15} = \varepsilon(\mathrm{erfc}, \mathbf{B}_3), \quad x \in \mathbf{B}_3 \cap S(2,53)\,.}$$

## 7.7 Fehlerschranke für erf($x$) in $\mathbf{B}_3 \cup \mathbf{B}_4$, d. h. für $x \ge 6$

In diesem Bereich wird erf($x$) durch 1 approximiert, d. h.

$$\mathrm{erf}(x) \approx 1\,.$$

Der relative Approximationsfehler $\varepsilon_{\mathrm{app}}(x)$ ist dann monoton fallend und es gilt

$$\varepsilon_{\mathrm{app}}(x) := \frac{1-\mathrm{erf}(x)}{\mathrm{erf}(x)} = \frac{1}{\mathrm{erf}(x)} - 1 = \frac{\mathrm{erfc}(x)}{1-\mathrm{erfc}(x)} \le \frac{\mathrm{erfc}(6)}{1-\mathrm{erfc}(6)},$$

wobei eine garantierte Oberschranke für $\mathrm{erfc}(6)/[1-\mathrm{erfc}(6)]$ mit dem Programm `upper_b3` von Seite 33 berechnet werden kann, wenn man dort lediglich die Anweisung `x:=11/intval(5)` ersetzt durch `x:=6` . Das Ergebnis lautet

$$\frac{\mathrm{erfc}(6)}{1-\mathrm{erfc}(6)} < 2.1520 \cdot 10^{-17} \quad \Longrightarrow \quad \varepsilon_{\mathrm{app}}(x) < 2.1520 \cdot 10^{-17} = \varepsilon(\mathrm{app}) \quad \Longrightarrow$$

$$\boxed{\widetilde{\mathrm{erf}}(x) = \mathrm{erf}(x)(1+\varepsilon_f); \quad |\varepsilon_f| \le 2.1520 \cdot 10^{-17} = \varepsilon(\mathrm{erf}, x \ge 6)\,.}$$

## 7.8 Approximation von erfc($x$) in $\mathbf{C} = (-\infty, 0]$

Wegen der Identität $\mathrm{erfc}(x) \equiv 1 - \mathrm{erf}(x)$ gilt in $\mathbf{C} = (-\infty, 0]$

$$\begin{aligned} \widetilde{\mathrm{erfc}}(x) &:= 1 \boxminus \widetilde{\mathrm{erf}}(x) = \mathrm{erfc}(x) \cdot (1+\varepsilon_f), \qquad \text{mit} \\ |\varepsilon_f| &\le 2 \cdot \varepsilon^* + (1 + 2 \cdot \varepsilon^*) \cdot \max_{-x \ge 0} \left[ \frac{1 \cdot 0 + \mathrm{erf}(-x) \cdot \varepsilon(\mathrm{erf})}{1 + \mathrm{erf}(-x)} \right]. \end{aligned}$$

Da $[\ldots]$ für $(-x) \to +\infty$ monoton wächst, folgt unmittelbar

$$|\varepsilon_f| \le 2 \cdot \varepsilon^* + (1 + 2 \cdot \varepsilon^*) \cdot 0.5 \cdot 2.7153 \cdot 10^{-15} < 1.5797 \cdot 10^{-15} \quad \Longrightarrow$$

$$\boxed{\widetilde{\mathrm{erfc}}(x) = \mathrm{erfc}(x)(1+\varepsilon_f); \quad |\varepsilon_f| \le 1.5797 \cdot 10^{-15} = \varepsilon(\mathrm{erfc}, \mathbf{C}), \; x \in \mathbf{C} \cap S(2,53)\,.}$$

## 7.9 Zusammenfassung der Ergebnisse für erf$(x)$ und erfc$(x)$

Damit ist für die Funktionen $\mathrm{erf}(x), \mathrm{erfc}(x)$ die Fehlerabschätzungen bzgl. des IEEE-double-Formats unter der Voraussetzung des Einsatzes einer nur **hochgenauen** Grundarithmetik (1 Ulp) in allen Teilbereichen durchgeführt. Die relativen Fehlerschranken sind die Maxima der jeweils berechneten Oberschranken:

$$\varepsilon(\mathrm{erf}) = 2.7153 \cdot 10^{-15}; \qquad |x| \in [1.97193 \cdot 10^{-308}, +\infty) \cap S(2,53)$$

$$\varepsilon(\mathrm{erfc}) = 5.8540 \cdot 10^{-15}; \qquad x \in (-\infty, 26.5432] \cap S(2,53)$$

Wertet man die Funktionen $\mathrm{erf}(x), \mathrm{erfc}(x)$ mit dem gleichen Algorithmus im IEEE double-Format in **maximal genauer** Arithmetik (d. h. mit Rundung zur nächstgelegenen Gleitkommazahl) aus, so findet man entsprechend die noch kleineren Fehlerschranken

$$\varepsilon(\mathrm{erf}) = 1.5643 \cdot 10^{-15}; \qquad |x| \in [1.97193 \cdot 10^{-308}, +\infty) \cap S(2,53)$$

$$\varepsilon(\mathrm{erfc}) = 3.2952 \cdot 10^{-15}; \qquad x \in (-\infty, 26.5432] \cap S(2,53)\,.$$

Die angegebenen numerischen Fehlerschranken beziehen sich auf Argumente im IEEE-double-Format. Die vorgestellte Methodik der Fehlerabschätzungen kann aber auch für andere Datenformate durchgeführt werden (siehe z. B. [5]).

# 8 Anhang A: Die Hilfsfunktion $e^{-x^2}$

Da die Funktion $f = e^{-x^2}$ z.B. bei der Implementierung der Fehlerfunktion und bei Dawsons Integral eine entscheidende Rolle spielt, wird für $f$ in diesem Abschnitt ein geeigneter Algorithmus mit einer entsprechende Fehlerabschätzung unter den folgenden Voraussetzungen angegeben:

- Die Grundoperationen im IEEE double-Format werden **hochgenau** ausgeführt, d.h.: $a \boxdot b = (a \cdot b)(1+\varepsilon), \quad |\varepsilon| \leq 2 \cdot \varepsilon^* = 2^{-52} = 2.220446\ldots \cdot 10^{-16}$
- Benutzt wird die schnelle, im IEEE double-Format nach dem Tabellenverfahren implementierte Exponentialfunktion (siehe [10, 11]), d. h.
  $\mathrm{EXP}(x) = e^x \cdot (1 + \varepsilon_{\mathrm{exp}}), \quad |\varepsilon_{\mathrm{exp}}| \leq 2.3580 \cdot 10^{-16} =: \varepsilon(\mathrm{exp})$

Um zu verhindern, daß die Funktionswerte $f(x)$ in den denormalisierten Zahlenbereich des IEEE double-Formats fallen, muß die Ungleichung

$$\begin{aligned} e^{-x^2} &\geq 2^{-1022} \quad \text{(kleinste positive normalisierte Zahl)} \quad \Longleftrightarrow \\ |x| &\leq \sqrt{1022 \cdot \ln(2)} = 26.615717509251260202\ldots \end{aligned}$$

erfüllt sein. Wegen $f(-x) \equiv f$ kann man sich auf nichtnegative Argumente $x$ beschränken, d. h. es wird nur

$$0 \leq x < 26.615717$$

betrachtet. Zunächst soll die Frage geklärt werden, warum $f = e^{-x^2}$ auf dem Rechner nicht einfach dadurch realisiert werden kann, daß man die schon implementierte Standard–Exponentialfunktion EXP(...) mit dem gestörten Argument $-x \boxdot x$ aufruft. Die entsprechende Fehlerabschätzung wird zeigen, daß dieses Verfahren ungeeignet ist.

Bedeutet EXP(x) die im IEEE -Format nach dem Tabellenverfahren implementierte Exponentialfunktion, so gilt zunächst

$$\mathrm{EXP}(x) = e^x \cdot (1 + \varepsilon_{\exp}), \qquad |\varepsilon_{\exp}| \leq 2.3580 \cdot 10^{-16} =: \varepsilon(\exp)$$

$$\mathrm{EXP}(-x \boxdot x) = e^{-x^2} \cdot (1 + \varepsilon_1), \qquad |\varepsilon_1| \leq \varepsilon(1) = \Gamma$$

Wegen

$$x \boxdot x = x^2 \cdot (1 + \varepsilon_x), \quad |\varepsilon_x| \leq 2 \cdot \varepsilon^* \leq 2.220447 \cdot 10^{-16} = \varepsilon(x),$$

$$x \in [0, 26.615717] \quad \Longrightarrow \quad x^2 \in [0, 708.397]$$

wird die Funktion EXP(...) mit gestörten Argumenten aufgerufen. Mit Hilfe des XSC-Programms `ErrBound` erhält man mit den Eingabedaten

| | |
|---|---|
| Intervall der exakten Argumente $[x1, x2] = \Gamma$ | $[0, 708.397]$ |
| Relative Schranke der gestörten Argumente | $2.220447 \cdot 10^{-16}$ |
| Rel. Fehlerschranke der Funktion im Raster S(B,k) $= \Gamma$ | $2.3580 \cdot 10^{-16}$ |

die unnötig große Schranke

$$\varepsilon(1) = 1.5754 \cdot 10^{-13}\,.$$

Eine um gut zwei Größenordnungen kleinere Fehlerschranke liefert der folgende Algorithmus, dessen Laufzeit im Vergleich zur schnellen Exponentialfunktion etwa doppelt so groß ist:

```
var x,m : real;
    z   : integer;
begin
  z := trunc(x);  { z: ganzzahliger Anteil des Arguments }
  m := x - z;     { m: zugehoeriger Nachkomma-Wert       }
  if m > 0.5 then
  begin
    z := z + 1;   { z = 0,1,2,....,27                    }
    m := m - 1;   { m <= 0.5;                 x = z + m; }
  end; ...
end.
```

Mit den so berechneten ganzzahligen Größen $m$ und $z$ wird dann $e^{-x^2}$ gemäß

$$e^{-x^2} = e^{-z^2} \cdot e^{-(2z)\cdot m} \cdot e^{-m^2} \tag{24}$$

berechnet. Die Konstanten $e^{-z^2}$ werden mit Hilfe des Moduls `mp_ari` berechnet und können für $z \in \{0, 1, \ldots, 26\}$ im IEEE-double-Format maximal genau gespeichert werden:

$$\widetilde{e^{-z^2}} = e^{-z^2} \cdot (1+\varepsilon); \qquad |\varepsilon| \le \varepsilon^* = 2^{-53} = 1.1102230\ldots \cdot 10^{-16}.$$

Wegen $e^{-27^2} = e^{-729} < 2^{-1022}$ kann die Konstante $e^{-27^2}$ im normalisierten Zahlenbereich des IEEE-double-Formats nicht mehr maximal genau gespeichert werden. Zerlegt man jedoch $e^{-729}$ in die zwei Faktoren:

$$e^{-729} = \left[2^{64} \cdot e^{-729}\right] \cdot 2^{-64},$$

so läßt sich der erste Faktor wegen $[2^{64} \cdot e^{-729}] > 2^{-1022}$ wieder maximal genau speichern, und die Multiplikation mit $2^{-64}$ kann sehr schnell und rundungsfehlerfrei durchgeführt werden. Die relative Fehlerschranke $\varepsilon(k)$ gilt damit für alle Faktoren $e^{-z^2}$.
Da der Exponent des zweiten Faktors in Gleichung (24) rundungsfehlerfrei berechnet wird, gilt

$$\mathrm{EXP}(-(z \boxplus z) \boxdot m) = \mathrm{EXP}(-(z+z)\cdot m) = e^{-(2z)\cdot m} \cdot (1+\varepsilon_{\exp})$$

$$|\varepsilon_{\exp}| \le 2.3580 \cdot 10^{-16} = \varepsilon(\exp).$$

Die Exponentialfunktion wird also bei dem hier vorgestellten Algorithmus mit dem **ungestörten** Argument $-(2z)\cdot m$ aufgerufen!
Der letzte Faktor $e^{-m^2}$ in (24) wird mit Hilfe der schnellen Exponentialfunktion ausgewertet, die mit dem gestörten Argument $-m \boxdot m$ aufgerufen wird, das jedoch jetzt auf das Intervall $[-0.25, 0]$ mit betragsmäßig sehr viel kleineren Werten beschränkt ist.

$$\mathrm{EXP}(-m \boxdot m) = e^{-m^2}(1+\varepsilon_2), \quad |\varepsilon_2| \le \varepsilon(2) = \Gamma$$

Mit Hilfe des Programms `ErrBound` ergibt sich mit dem nun wesentlich kleineren Argumentintervall

$$\text{Intervall der exakten Argumente } [x1, x2] = \Gamma \Big| \; [-0.25,\, 0]$$

(die weiteren Eingabedaten stimmen mit denen bei der Berechnung von $\varepsilon(1)$ angegebenen überein) die gesuchte Oberschranke für den Betrag des relativen Berechnungsfehlers $\varepsilon_2$ zu

$$\varepsilon(2) = 2.9132 \cdot 10^{-16}\,.$$

Damit sind die Fehlerschranken der drei Faktoren in (24) abgeschätzt und man erhält mit der Darstellung

$$\widetilde{e^{-x^2}} = e^{-z^2}(1+\varepsilon) \cdot e^{-(2z)\cdot m}(1+\varepsilon_{exp}) \cdot e^{-m^2}(1+\varepsilon_2) \cdot (1+2\varepsilon)^2 = e^{-x^2}(1+\varepsilon_3)$$

für $|\varepsilon_3|$ wieder mit Hilfe des Programs `ErrBound` unter der Annahme einer **hochgenauen** Arithmetik als Oberschranke für den Gesamtfehler

$$\boxed{\widetilde{e^{-x^2}} = e^{-x^2}(1+\varepsilon_3);\;\; |\varepsilon_3| \le 1.0823 \cdot 10^{-15},\;\; |x| \in [0,\, 26.615717] \cap S(2,53)\,.}$$

Der Vergleich mit der Fehlerschranke $\varepsilon(1) = 1.5754 \cdot 10^{-13}$ zeigt, daß an Stelle von nur etwa 13 Ziffern die Funktionswerte $e^{-x^2}$ jetzt mit mindestens 15 korrekten Dezimalziffern berechnet werden können.

Unter der Annahme einer **maximal genauen Arithmetik** erhält man bei sonst gleichem Algorithmus die nochmals verbesserte relative Gesamtfehlerschranke

$$\boxed{\widetilde{e^{-x^2}} = e^{-x^2}(1+\varepsilon_3);\;\; |\varepsilon_3| \le 8.3243 \cdot 10^{-16},\;\; |x| \in [0,\, 26.615717] \cap S(2,53)\,.}$$

# 9 Anhang B: Ein einfaches Testprogramm, numerische Resultate

Das folgende Programm verwendet die Beziehung

$$\mathrm{erf}(x) + \mathrm{erfc}(x) - 1 = 0$$

für einen einfachen Test. Bei der intervallmäßigen Auswertung der linken Seite muß sich jeweils ein Intervall ergeben, das die Null enthält.

Der Benutzer muß nur das Modul `erf_mod` mit dem Kommando `use erf_mod` in sein PASCAL-XSC Programm einbinden, um sowohl die erf als auch die erfc Funktion für Intervallargumente zur Verfügung zu haben.

```
PROGRAM erf_test;
USE i_ari, erf_mod;
VAR x, fx: interval;
BEGIN
  writeln;
  writeln('           ***********************************************');
  writeln('           *    Berechnung von erf(x) und erfc(x)     *');
  writeln('           ***********************************************');
  writeln;
  writeln('Programmabbruch mit <Ctrl> <C> ');
  REPEAT
    writeln;
    write('x = [xlb, xub] = ?  '); read(x); writeln;
    writeln('Argumentintervall:', x); writeln;
    fx := erf(x);
    writeln('erf(x)  = [', fx.inf:23:0:-1, ' , ', fx.sup:23:0:+1, ' ]');
    fx := erfc(x);
    writeln('erfc(x) = [', fx.inf:23:0:-1, ' , ', fx.sup:23:0:+1, ' ]');
    writeln('erf(x) + erfc(x) - 1: ', erf(x) + erfc(x) - 1 );
  UNTIL FALSE

END.
```

Programmausgabe zum Testprogramm:

```
            ***********************************************
            *    Berechnung  von erf(x) + erfc(x) - 1      *
            ***********************************************

Programmabbruch mit <Ctrl> <C>

x = [xlb, xub] = ?
Argumentintervall:[  1.000000000000000E+000,  1.000000000000000E+000 ]

erf(x)  = [ 8.427007929497132E-001 ,  8.427007929497166E-001 ]
erfc(x) = [ 1.572992070502843E-001 ,  1.572992070502860E-001 ]
erf(x) + erfc(x) - 1: [               -2.6E-015,               2.7E-015 ]


x = [xlb, xub] = ?
Argumentintervall:[  5.000000000000000E+000,  5.000000000000000E+000 ]

erf(x)  = [ 9.999999999984608E-001 ,  9.999999999984643E-001 ]
erfc(x) = [ 1.537459794428029E-012 ,  1.537459794428042E-012 ]
erf(x) + erfc(x) - 1: [               -1.7E-015,               1.8E-015 ]

x = [xlb, xub] = ?
Argumentintervall:[ -1.000000000000000E+000, -1.000000000000000E+000 ]

erf(x)  = [-8.427007929497166E-001 , -8.427007929497132E-001 ]
erfc(x) = [ 1.842700792949708E+000 ,  1.842700792949722E+000 ]
erf(x) + erfc(x) - 1: [               -8.3E-015,               8.5E-015 ]

x = [xlb, xub] = ?
Argumentintervall:[ -2.000000000000000E+000, -2.000000000000000E+000 ]

erf(x)  = [-9.953222650189544E-001 , -9.953222650189510E-001 ]
erfc(x) = [ 1.995322265018946E+000 ,  1.995322265018960E+000 ]
erf(x) + erfc(x) - 1: [               -8.3E-015,               8.5E-015 ]
```

```
x = [xlb, xub] = ?
Argumentintervall:[  2.000000000000000E+002,  2.000000000000000E+002 ]

erf(x)  = [ 9.999999999999967E-001 ,  1.000000000000000E+000 ]
erfc(x) = [ 0.000000000000000E+000 ,  4.450147717014403E-308 ]
erf(x) + erfc(x) - 1: [               -3.3E-015,                2.3E-016 ]
```

Die Ergebnisse spiegeln im wesentlichen die Größenordnung der Fehlerschranken der Implementierungen von erf und erfc wieder. Auch liegt in allen getesteten Fällen die Null im berechneten Ergebnisintervall.

# Literatur


[1] G. Alefeld, J. Herzberger: *Einführung in die Intervallrechnung.* Reihe Informatik/12; BI, 1974.

[2] M. Abramowitz and I.A. Stegun: – *Handbook of Mathematical Functions, Graphs, and Mathematical Tables.* Dover Publications, INC., NEW YORK.

[3] F. Blomquist: *Automatische a priori Fehlerabschätzungen.* Inst. f. Angewandte Mathematik, Universität Karlsruhe, 1995.

[4] F. Blomquist: *Mathematische Funktionen für Intervallargumente.* Interner Abschlussbericht[1] eines gleichnamigen Projektes, Inst. für Angewandte Mathematik, Universität Karlsruhe, 1997.

[5] F. Blomquist: *Dezimalversion von PASCAL-XSC.* Inst. für Angewandte Mathematik, Universität Karlsruhe, erscheint 1998.

[6] K. D. Braune: *Hochgenaue Standardfunktionen für reelle und komplexe Punkte und Intervalle in beliebigen Gleitpunktrastern.* Dissertation , Universität Karlsruhe 1987.

[7] B.D. Fried and S.D. Conte: *The plasma dispersion function.* Academic Press, New York, N.Y. and London, England, 1961.

[8] W. Gautschi: *Error function and Fresnel integrals.* Chapter 7 in [2].

[9] R. Hammer, M. Hocks, U. Kulisch, D. Ratz: *Numerical Toolbox for Verified Computing I.* Springer Series in Computational Mathematics 21, 1993.

[10] W. Hofschuster, W. und Krämer: *Ein rechnergestützter Fehlerkalkül mit Anwendung auf ein genaues Tabellenverfahren.* Preprint 96/5 des IWRMM, Universität Karlsruhe, 35 Seiten, 1996.

[11] W. Hofschuster, W. Krämer: *A Computer Oriented Approach to Get Sharp Reliable Error Bounds*, *Reliable Computing 3*, pp. 239-248, 1997.


[1] Dieser Bericht kann am IWRMM eingesehen werden.

[12] W. Krämer: *Inverse Standardfunktionen für reelle und komplexe Intervallargumente mit a priori Fehlerabschätzungen für beliebige Datenformate.* Dissertation, Universität Karlsruhe, 1987.

[13] W. Krämer: *Eine portable Langzahl- und Langzahlintervallarithmetik mit Anwendungen*, Z. angew. Math. Mech. 73, 1992.

[14] W. Krämer: *Sichere und genaue Abschätzung des Approximationsfehlers bei rationalen Approximationen.* Forschungsschwerpunkt Computerarithmetik, Intervallrechnung und Numerische Algorithmen mit Ergebnisverifikation, Bericht 3/1996, Karlsruhe, 1996.

[15] W. Krämer: *Eine Fehlerfaktorarithmetik für zuverlässige a priori Fehlerabschätzungen.* Forschungsschwerpunkt Computerarithmetik, Intervallrechnung und Numerische Algorithmen mit Ergebnisverifikation, Bericht 5/1997, 21 Seiten, Karlsruhe, 1997.

[16] W. Krämer: *A priori Worst Case Error Bounds for Floating-Point Computations.* Proceedings of the 13th IEEE Symp. on Computer Arithmetic, Asilomar, California, pp. 64-71, 1997.

[17] W. Krämer: *Constructive Error Analysis*, accepted for publication in: Journal of Universal Computer Science (JUCS).

[18] B. Lohmander, S. Rittsten: *Table of the Function* $y = e^{-x^2} \int_0^x e^{t^2} dt$, Kungl. Fysiogr. Sällsk. i. Lund Förh., V 28, 1958, pp. 45–52.

[19] Y.L. Luke: *The Special Functions and their Approximations.* Volume I, Academic Press, NEW YORK and London, 1969.

[20] Y.L. Luke: *The Special Functions and their Approximations.* Volume II; Academic Press, NEW YORK and London, 1969.

[21] Y.L. Luke. *Algorithms for the Computation of mathematical Functions* Academic Press, NEW YORK SAN FRANCISCO LONDON, 1977.

[22] Y.L. Luke: *Mathematical Functions and their Approximations* Academic Press, NEW YORK SAN FRANCISCO LONDON, 1975.

[23] *Algebrasystem Mathematica.* Wolfram Research, Inc., Champaign, Illinois.

[24] W.L. Miller and A.R. Gordon: *Numerical evaluation of infinit series.* Jn. Phys. Chem., V. 35, 1931, especially part V, p. 2856-2857, 2860-2865.

[25] F. Oberhettinger: *Tabellen zur Fourier Transformation.*
Springer, Berlin, 1957.

[26] PASCAL–XSC: A PASCAL Extension for Scientific Computation and Numerical Data Processing. Numeric Software GmbH, D–76492 Baden–Baden, Germany.

[27] J.B. Rosser: *Theorie and Application of* $\int_0^z e^{-x^2}dx$ *and* $\int_0^z e^{-p^2y^2}dy \int_0^y e^{-x^2}dx$. *Part I.* Methods of Computation, NEW YORK, 1948.

[28] H.E. Salzer: *Formulas for calculating the error function of a complex variable.* Math. Tables and Other Aids to Computation 5, 67-70, (1951).

[29] E.C. Titchmarsh: *Introduction to the Theory of Fourier Integrals.* Oxford, 1937, p. 60-64.

[30] *IEEE Standard for Binary Floating-Point Arithmetic, ANSI–IEEE Standard 754-1985, 1985.*

**In dieser Reihe sind bisher die folgenden Arbeiten erschienen:**

Nr. 93/1: G. Aumann, K. Bentz: Geometrische Stetigkeit beliebiger Ordnung zwischen Tensor-Produkt-Bézier-Flächen

Nr. 93/2: G. Alefeld, G. Mayer: A Computer Aided Existence and Uniqueness Proof for an Inverse Matrix Eigenvalue Problem

Nr. 93/3: B. Weber: Symbolische Programmierung in der Mehrkörperdynamik

Nr. 93/4: R. Rihm: Über Einschließungsverfahren für gewöhnliche Anfangswertprobleme und ihre Anwendung auf Differentialgleichungen mit unstetiger rechter Seite

Nr. 93/5: J. Wittenburg: Explizite Lösungen für lineare Gleichungssysteme mit tridiagonalen Koeffizientenmatrizen. Anwendungen in der Mechanik

Nr. 93/6: N. Henze, B. Klar: Goodness–of–Fit Testing for a Space–Time Model for Daily Rainfall

Nr. 93/7: K. Schweizerhof, J. Riccius, M. Baumann: Verbesserung von Finite Element Berechnungen durch Adaptivität und Netzglättung am Beispiel ebener und gekrümmter Flächentragwerke

Nr. 93/8: G. Starke: Subspace Orthogonalization for Substructuring Preconditioners for Nonselfadjoint Elliptic Problems

Nr. 93/9: N. Henze, B. Klar: Empirical Distribution Function Tests for the Generalized Poisson Model

Nr. 94/1: G. Aumann: Geometric Continuity of Parametric Curves and Surfaces

Nr. 94/2: T. Dehn, M. Eiermann, K. Giebermann, V. Sperling: Structured Sparse Matrix-Vector Multiplication on Massively Parallel Architectures

Nr. 94/3: W. Krämer: Bericht über die Begutachtung des IWRMM im Dezember 1993

Nr. 95/1: L. Kobbelt: Interpolatory Refinement is Low Pass Filtering

Nr. 95/2: M. Paluszny, H. Prautzsch, M. Schäfer: Corner cutting and interpolatory refinement

Nr. 95/3: B. Klar: Analysis of and Goodness of Fit Testing for a Flexible Discrete Time Failure Model

Nr. 95/4: P. Vielsack: Regularisierung von Haftkräften bei Coulombscher Reibung

Nr. 95/5: P. Vielsack, M. Storz: Bifurcation of Motion in a Technical System with Stick-Slip and Impact

Nr. 95/6: M. Brühl: A Curve Tracing Algorithm for Computing the Pseudospectrum

Nr. 95/7: J. Riccius, K. Schweizerhof, M. Baumann: On the treatment of shell intersections in adaptive finite element analysis and combination with mesh smoothing

Nr. 96/1: M. Dormanns, H.-U. Heiß: Nutzung von Asynchronität bei iterativen Gleichungslösern auf Multirechnersystemen

Nr. 96/2: P. Vielsack, J. Kirillowa: Nichteindeutigkeit der Bewegungen eines Reibschwingers mit Selbsterregung

Nr. 96/3: L. Kobbelt, T. Hesse, H. Prautzsch, K. Schweizerhof: Diskrete Freiformflächenerzeugung für FEM-Anwendungen

Nr. 96/4: M. Brühl, M. Hanke, H. Wanzki: Ein Rekonstruktionsverfahren für die elektrische Impedanztomographie

Nr. 96/5 : W. Hofschuster, W. Krämer: Ein rechnergestütztes Fehlerkalkül mit Anwendung auf ein genaues Tabellenverfahren

Nr. 96/6: W. Niethammer, W. Krämer (Herausgeber): Tagungsband zum Workshop „Wissenschaftliches Rechnen in den Ingenieurwissenschaften“

Nr. 96/7: G. Freimann: FAS-Verfahren zur Lösung strukturmechanischer Probleme

Nr. 97/1: P. Vielsack, A. Hartung: Orbitale Stabilität von Bewegungen mit Pausen bei Einwirkung permanenter Störungen

Nr. 97/2: J. G. Schmidt, G. Starke: Coarse Space Orthogonalization for Indefinite Linear Systems of Equations Arising in Geometrically Nonlinear Elasticity

Nr. 97/3: F. Blomquist, W. Krämer: Algorithmen mit Fehlerschranken für die Fehler- und die komplementäre Fehlerfunktion

Weitere Arbeiten sind in Vorbereitung.